

Instituto Superior de Engenharia do Porto

# **Sistema Pericial para o Estudo de Composição de Betões**

André Teixeira David

Dissertação para obtenção do Grau de Mestre em

Engenharia Informática

Área de Especialização em Tecnologias do Conhecimento e Decisão

Orientador: Professor Luiz Felipe Faria

Co-orientador: Eng.º Jorge da Fonseca Lourenço

Júri:

Presidente: Doutora Maria de Fátima Coutinho Rodrigues, Professora Coordenadora, ISEP

Vogais: Doutora Isabel Cecília Correia da Silva Praça Gomes Pereira, Professora Adjunta, ISEP

Doutor Luiz Felipe Rocha de Faria, Professor Adjunto, ISEP

Eng.º Jorge da Fonseca Lourenço, Professor Adjunto, ISEC

Porto, Outubro de 2012



# Agradecimentos

Gostaria de agradecer ao Professor Luiz Felipe Faria por ter viabilizado este projecto e pela sua preciosa ajuda e disponibilidade na progressão do trabalho.

Tenho também uma enorme dívida de gratidão para com o Eng.<sup>o</sup> Jorge Lourenço, elemento basilar deste trabalho, que demonstrou uma imensa paciência e compreensão, e cuja orientação e conhecimentos técnicos foram decisivos.

Ao Zé David, pelos seus ensinamentos e apoio neste projecto.

Por último, uma palavra de agradecimento aos meus pais, sem quem nunca teria chegado aqui, e à Evi, por todo o seu apoio nos momentos mais importantes.



# Resumo

O betão é o material de construção feito pelo Homem mais utilizado no mundo. A sua composição é um processo complexo que exige um conhecimento teórico sólido e muita experiência prática, pelo que poucas pessoas estão habilitadas para o fazer e são muito requisitadas. No entanto não existe muita oferta actual de software que contemple alguns dos aspectos importantes da composição do betão, nomeadamente para o contexto europeu.

Nesse sentido, foi desenvolvido um sistema de apoio à decisão chamado *Betacomp*, baseado num sistema pericial, para realizar estudos de composição de betão. Este contempla as normas legais portuguesas e europeias, e a partir da especificação do betão apresenta toda a informação necessária para se produzir um ensaio de betão. A aquisição do conhecimento necessário ao sistema contou com a colaboração de um especialista com longa e comprovada experiência na área da formulação e produção do betão, tendo sido construída uma base de conhecimento baseada em regras de produção no formato *drl* (*Drools Rule Language*). O desenvolvimento foi realizado na plataforma *Drools.net*, em *C#* e *VB.net*. O *Betacomp* suporta os tipos de betão mais comuns, assim como adições e adjuvantes, sendo aplicável numa grande parte dos cenários de obra. Tem a funcionalidade de fornecer explicações sobre as suas decisões ao utilizador, auxiliando a perceber as conclusões atingidas e simultaneamente pode funcionar como uma ferramenta pedagógica.

A sua abordagem é bastante pragmática e de certo modo inovadora, tendo em conta parâmetros novos, que habitualmente não são considerados neste tipo de software. Um deles é o nível do *controlo de qualidade do produtor de betão*, sendo feito um ajuste de compensação à resistência do betão a cumprir, proporcional à qualidade do produtor. No caso dos produtores de betão, permite que indiquem os constituintes que já possuem para os poderem aproveitar (caso não haja impedimentos técnicos), uma prática muito comum e que permitirá eventualmente uma aceitação maior da aplicação, dado que reflecte a forma habitual de agir nos produtores.

**Palavras-chave:** sistema pericial, drools, .NET, estudo de composição, betão, Betacomp



# Abstract

Concrete is the most widely used man made construction material in the World. Concrete mix design is a complex process that requires a solid theoretical knowledge and field experience, hence the reduced number of people able to perform it and their limited available time. Nonetheless, software offer for concrete mix design is rather small, especially regarding the European context.

Therefore, a new Decision Support System named *Betacomp* was developed in order to fill this gap. Based on an Expert System, it produces the necessary concrete mix design information for creating trial batches from the concrete specification, whilst enforcing the mandatory legal concrete standards for Europe.

The necessary knowledge acquisition derives from the expertise of a well renowned specialist in the field of concrete mix design and provides the necessary information for the production rules' Knowledge Base, built on the *Drools Rule Language (drl)* format. Its was developed in the *Drools.net* platform, *C#* and *VB.NET*.

*Betacomp* supports the most common types of concrete, as well as concrete admixtures and additives, thus making its usage possible in most construction sites. It also deploys an Explanation Module for explaining the decisions made by the system to the user, therefore making them more understandable and simultaneously working as a learning tool.

It has a very pragmatic and even innovative approach towards concrete mix design, considering parameters not usually found in this kind of software. One of the new features is the *Quality Control of the Producer*, which considers adjustments to the concrete strength according to the quality control enforced by the producer. Especially if the user is a concrete producer, he can also select beforehand the preferred constituents he already possesses in order to minimize the concrete cost (if there are no technical hindrances), a common practice among the industry and that hopefully will allow for a bigger acceptance of this software.

**Keywords:** expert system, drools, .NET, mix design, concrete, Betacomp



# Conteúdo

<b>Agradecimentos</b>	<b>iii</b>
<b>Resumo</b>	<b>v</b>
<b>Abstract</b>	<b>vii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Enquadramento . . . . .	1
1.2 Definição do problema e âmbito do trabalho . . . . .	1
1.3 Objectivos . . . . .	2
1.4 Métodos e tecnologias utilizadas . . . . .	2
1.5 Organização da dissertação . . . . .	3
<b>2 Contexto</b>	<b>5</b>
2.1 O betão . . . . .	5
2.2 O problema da composição do betão . . . . .	6
2.3 O cálculo da composição do betão . . . . .	8
2.3.1 A composição do betão . . . . .	8
2.3.2 Metodologia para o cálculo da composição do betão . . . . .	8
2.4 Os Sistemas Periciais . . . . .	10
2.4.1 A Inteligência Artificial . . . . .	11
2.4.2 Descrição geral de um Sistema Pericial . . . . .	11
2.4.3 Tópicos fundamentais dos Sistemas Periciais . . . . .	14
<b>3 Estado da Arte</b>	<b>19</b>
3.1 Ferramentas para desenvolvimento de Sistemas Periciais . . . . .	19
3.2 Soluções existentes para calcular a composição do betão . . . . .	25
3.2.1 Sistemas Periciais para a definição da composição do betão . . . . .	26

3.2.2	Outros Sistemas Periciais para betão . . . . .	36
3.2.3	Sistemas de outra natureza utilizados para cálculo da composição do betão . . . . .	39
<b>4</b>	<b>Definição do Problema</b>	<b>45</b>
4.1	Identificação do problema . . . . .	45
4.2	Análise comparativa dos sistemas analisados . . . . .	45
4.3	Necessidade de soluções . . . . .	48
4.4	Objectivos e relevância da aplicação . . . . .	50
<b>5</b>	<b>Sistema Pericial Desenvolvido</b>	<b>53</b>
5.1	Tecnologias relevantes . . . . .	53
5.1.1	A escolha de um paradigma . . . . .	53
5.1.2	Ferramentas de desenvolvimento consideradas . . . . .	54
5.1.3	Outras ferramentas utilizadas . . . . .	56
5.2	Arquitectura do sistema . . . . .	57
5.2.1	Módulos do <i>Betacomp</i> . . . . .	57
5.2.2	Interligação dos módulos . . . . .	59
5.2.3	Fluxo de invocação . . . . .	59
5.3	Aquisição do conhecimento . . . . .	61
5.3.1	O conhecimento pericial . . . . .	61
5.3.2	As normas legais a respeitar . . . . .	62
5.3.3	Método de aquisição do conhecimento . . . . .	62
5.4	Representação do conhecimento . . . . .	62
5.4.1	Anatomia de uma regra . . . . .	62
5.4.2	Base de conhecimento . . . . .	64
5.5	Raciocínio . . . . .	65
5.5.1	Resolução de conflitos e proeminência . . . . .	66
5.5.2	Prevenção de ciclos recursivos . . . . .	66
5.5.3	Agrupamento de regras . . . . .	67
5.6	Utilização do <i>Betacomp</i> . . . . .	67
5.6.1	Introdução dos <i>inputs</i> pelo utilizador . . . . .	68
5.6.2	Inferência . . . . .	70
5.6.3	Módulo de cálculo de composição de betão . . . . .	72
5.7	Explicação das soluções . . . . .	75
5.8	Validação do sistema . . . . .	76

<b>6 Conclusões</b>	<b>79</b>
6.1 Resultados obtidos pela solução desenvolvida . . . . .	79
6.2 Contribuições da solução desenvolvida . . . . .	80
6.3 Desenvolvimentos futuros . . . . .	81
<b>Bibliografia</b>	<b>83</b>
<b>Anexos</b>	<b>89</b>
<b>A Questões de ordem prática e tomada de opções</b>	<b>89</b>
<b>B Regras de produção do <i>Betacomp</i></b>	<b>91</b>
<b>C Listagem parcial das explicações do sistema</b>	<b>105</b>



# Lista de Figuras

2.1	Exemplos do que o betão permitiu edificar . . . . .	6
2.2	Uma central de betão pronto . . . . .	7
2.3	Curva granulométrica de referência e curva da mistura real . . . . .	9
2.4	Fluxograma do processo da composição de betões de granulometria óptima . . . . .	10
2.7	A representação de um <i>frame</i> . . . . .	16
2.8	Uma rede semântica representada por um grafo . . . . .	17
3.1	O ambiente de programação do CLIPS 6.2 no S.O. <i>Windows 7</i> . . . . .	20
3.2	LPA Win-Prolog v4.920 . . . . .	21
3.3	O Eclipse IDE com o plugin de suporte ao Drools . . . . .	22
3.4	O Exsys Corvid Expert System Development Tool da EXSYS . . . . .	24
3.5	Fluxograma de ajuste da mistura no HPCMIX . . . . .	30
3.6	Composição do betão no HPCMIX para o primeiro ensaio . . . . .	31
3.7	Janela do módulo de <i>Mix Performance Adjustment</i> do HPCMIX . . . . .	31
3.8	A arquitectura do sistema de auto-aquisição de conhecimento . . . . .	33
3.9	Janela do <i>Mix Design</i> do ESBAC mostrando o fluxograma do mesmo . . . . .	34
3.10	Os inputs necessários para o primeiro ensaio de laboratório . . . . .	35
3.11	Janela do ESBAC com a composição do betão e dados do primeiro ensaio . . . . .	35
3.12	Uma explicação do ESBAC durante a entrada de dados . . . . .	36
3.13	Lista de defeitos do BRIDGE-1 . . . . .	37
3.14	Causas prováveis de um certo defeito . . . . .	38
3.15	Exemplo de uma regra da base de conhecimento do BRIDGE . . . . .	38
3.16	Diagnóstico, explicação e recomendações na interface do <i>Matlab</i> . . . . .	40
3.17	O <i>User's Guide</i> no ambiente <i>Matlab</i> . . . . .	40
3.18	Fluxograma do diagnóstico para ajustes na mistura do betão . . . . .	41
3.19	Ecrã inicial do Matwin . . . . .	43
3.20	Ecrã inicial do Sikacomp 2.5 . . . . .	44

5.1	A interface do sistema pericial - projecto <i>Betacomp_SP</i> em <i>C#</i> . . . . .	58
5.2	O ecrã principal do módulo de cálculo de composição - projecto <i>BetacompCalc</i> em <i>VB.NET</i>	58
5.3	Integração dos componentes e principais classes do <i>Betacomp</i> . . . . .	60
5.4	O Laboratório de Eng. Civil do ISEC, local das sessões de aquisição de conhecimento . .	61
5.5	A execução de regras no Drools . . . . .	65
5.6	Fluxograma representativo do raciocínio do sistema pericial desenvolvido . . . . .	69
5.7	Resultados na interface do sistema pericial - <i>Betacomp</i> . . . . .	70
5.8	O ecrã inicial do módulo de cálculo de composição do <i>Betacomp</i> . . . . .	73
5.9	A análise granulométrica no <i>Betacomp</i> . . . . .	73
5.10	A análise gráfica das curvas granulométricas e dos agregados no <i>Betacomp</i> . . . . .	74
5.11	A análise de custos no <i>Betacomp</i> . . . . .	75
5.12	A <i>Explicação das decisões tomadas e outras informações</i> no <i>Betacomp</i> . . . . .	76
5.13	A <i>Consola de Sistema</i> no <i>Betacomp</i> . . . . .	76

# Lista de Tabelas

4.1	Comparação entre os sistemas para estudos de composição de betão analisados . . . . .	47
5.1	Comparação entre algumas das principais ferramentas de desenvolvimento de sistemas periciais . . . . .	55



# Capítulo 1

## Introdução

*“Opportunity is missed by most people because it is dressed in overalls and looks like work.”*

—Thomas A. Edison

### 1.1 Enquadramento

No âmbito de uma colaboração com a sucursal de uma empresa multinacional suíça da indústria química para a construção - a SIKA Portugal S.A. - foi desenvolvida uma aplicação pelo proponente, o *Sikacomp 2.5*, para o estudo de composição de betões. Todavia, o trabalho realizado no âmbito desta dissertação é de natureza independente, sem colaboração com esta entidade. O principal público alvo deste software eram as empresas de betão pronto, pré-fabricadoras de betão, engenheiros civis e operadores de laboratório, entre outros.

À altura da escrita desta dissertação, o *Sikacomp 2.5* é utilizado para estudos de composição de betão em obras em Portugal e Angola. No entanto, mesmo com este, a introdução dos diversos parâmetros de entrada necessários é um processo moroso e com várias implicações. Existem muitas correlações entre as variáveis e várias restrições nos valores a introduzir, que não obstante serem efectivamente controlados pelo programa, traduzem-se em alguma complexidade, que sem os conhecimentos periciais e experiência necessários, tornam a sua utilização mais restrita ou menos acessível.

### 1.2 Definição do problema e âmbito do trabalho

A composição do betão é um processo complexo que exige um conhecimento teórico sólido e muita experiência prática. Pouco pessoas estão habilitadas para o fazer e estas são peritos extremamente requisitados, o que faz com que o seu tempo seja bastante limitado e obrigando muitas vezes a deslocações

dispendiosas a locais de obra. Neste contexto, seria útil ter disponível uma ferramenta que permitisse libertar parte do tempo destes profissionais e que idealmente encerrasse uma vertente pedagógica que auxiliasse na aprendizagem, através da realização de estudos de composição de betão acompanhados de explicações das decisões tomadas.

A ideia inerente não é tomar o lugar destes peritos, o que muito dificilmente seria possível dados os constrangimentos práticos deste tipo de trabalho, mas antes o de os substituir *parcialmente* através da aquisição do conhecimento relevante por detrás das suas tomadas de decisão, conseguindo deste modo consideráveis aumentos de desempenho na produção do betão.

Foi esse o trabalho que se procurou desenvolver na elaboração desta tese, o desenvolvimento de um sistema pericial (SP) que auxiliasse e tomasse decisões na criação de estudos de composição de betão. Os tipos de betão contemplados são os normais e os auto-compactáveis.

### 1.3 Objectivos

Na sequência desta lógica, pretendeu-se desenvolver um Sistema de Apoio à Decisão (SAD) que prestasse apoio no estudo de composição de betões. Este SAD tem por base um sistema pericial e é uma evolução da aplicação já desenvolvida e atrás referida (*Sikacomp 2.5*). A construção da respectiva base de conhecimento contou com a preciosa colaboração de um especialista com longa e comprovada experiência na área da engenharia civil, mais concretamente na formulação e produção do betão.

Os objectivos principais a alcançar seriam, portanto, dotar a nova aplicação de uma base de conhecimento com a informação relevante e o raciocínio necessário para atingir o propósito de:

- com a assistência do programa, substituir parcialmente o perito humano, ao permitir que alguém atinja níveis de desempenho próximos aos de um especialista;
- no caso de o utilizador ser já um perito, melhorar a sua produtividade, sendo possível produzir resultados mais rapidamente;
- dotar a ferramenta de capacidade pedagógica, com a facilidade de criação de diferentes cenários alternativos para análise e experimentação, assim como explicações sobre as opções tomadas pelo sistema.

### 1.4 Métodos e tecnologias utilizadas

O primeiro passo, paralelamente à escolha de ferramentas a utilizar e demais planeamento, foi a aquisição do conhecimento necessário para a construção da base de conhecimento. Para tal foram realizadas várias sessões com um perito da área, onde progressivamente foi sendo realizada a extracção de

conhecimento importante e necessário à criação do sistema pericial, tentando compreender e traduzir a lógica inerente ao processo de composição de betão.

Relativamente às tecnologias utilizadas, após um processo inicial de comparação entre a oferta disponível, as ferramentas escolhidas para a criação do SP foram o *Microsoft Visual Studio 2008*, sendo utilizadas as linguagens *Visual Basic* (VB.NET) e *C#* (C Sharp). Para o motor de inferência e base de conhecimento foi utilizado o *Drools.NET*, uma versão da *Sourceforge* para a plataforma .NET da ferramenta *Drools*, a versão gratuita do *JBoss Rules* da Red Hat. Aqui foi construída uma versão do conhecimento adquirido do especialista em betão, na forma de regras de produção ou *business rules*.

Após a recolha das entradas do utilizador, o sistema pericial entra em funcionamento e despoleta as regras relevantes e produz resultados, ou seja, mostra quais os constituintes do betão a produzir e as suas quantidades, assim como algumas relações entre os mesmos e uma breve explicação. A validação destes resultados é realizada não só pelas próprias regras, o conhecimento do perito, mas também pelas normas legais em vigor em Portugal e no resto da Europa, obrigatórias para a produção de betão, as quais impõem restrições de diversas naturezas e que estão representadas nas regras *Drools* do sistema.

Desenvolveu-se portanto uma aplicação que integra diferentes paradigmas e diferentes ambientes de desenvolvimento. No *Visual Studio* foi desenhada a interface e gerada a funcionalidade necessária para a interacção com o utilizador, utilizando programação procedimental característica de uma linguagem visual tradicional, ou seja, rotinas, subrotinas, métodos, eventos e funções.

A base de conhecimento por sua vez é constituída na sua essência por um conjunto de regras simples e de fácil compreensão, e a sua produção controlada por um motor de inferência de acordo com o paradigma declarativo do Drools, um *Business Rules Management System*. O desenvolvimento do código *Drools* foi realizado no Eclipse IDE (Integrated Development Environment) na versão *Europa*, com a ajuda de um *plugin* da Red Hat para suporte a este *framework*.

## 1.5 Organização da dissertação

No primeiro capítulo - “Introdução” - (o actual), é apresentada a estrutura geral da dissertação, o problema a abordar, quais os objectivos a atingir e qual a abordagem escolhida.

O segundo capítulo, “Contexto”, situa o tema da dissertação, tentando transmitir ao leitor uma descrição pragmática de o que é um betão e no que consiste a ciência por trás da composição deste material. Dado que a abordagem seleccionada para endereçar esta questão foi a criação de um sistema pericial, é também descrita a essência deste tipo de sistemas.

O terceiro capítulo, “Estado da Arte”, é uma sùmula do que de mais relevante foi produzido na área da composição de betão, com especial destaque para os sistemas periciais para a composição de betão, mas apresentando também algumas alternativas como software comercial tradicional e outros tipos de soluções.

No quarto capítulo, “Definição do Problema”, é abordado o problema da especificação do betão em mais detalhe e são apresentados argumentos para fundamentar a necessidade de novas soluções nesta área. Aqui são expostos os objectivos concretos da aplicação a desenvolver e a sua relevância.

O quinto capítulo, “Sistema Pericial Desenvolvido”, começa por apresentar em pormenor a solução proposta para o problema em causa e quais as ferramentas utilizadas para o desenvolvimento do sistema pericial, justificando as opções tomadas. Esta é uma das secções mais importantes da dissertação, onde é também exposta a metodologia utilizada, como foi realizada a aquisição e a representação do conhecimento e qual o raciocínio posto em prática pelo sistema.

O sexto e final capítulo “Conclusões”, apresenta os resultados obtidos pela solução desenvolvida e quais as contribuições mais importantes que daí advieram. É feito um balanço do trabalho executado e são expostos alguns possíveis desenvolvimentos que poderão vir a ser realizados no futuro.

## Capítulo 2

# Contexto

*“We rarely recognize how wonderful it is that a person can traverse an entire lifetime without making a single really serious mistake — like putting a fork in one’s eye or using a window instead of a door.”*

— Marvin Minsky, *The Society of Mind* (1987).

### 2.1 O betão

O Betão é um dos materiais de construção mais utilizados em Engenharia Civil. O seu uso remonta à Roma antiga, onde foi utilizado na edificação de diversas estruturas, como o *Pantheon* e o *Colosseum*, abrindo espaço a uma verdadeira revolução nos materiais de construção, ao permitir novas formas e o desenho de edificações mais complexas. A construção começava então a ficar livre de muitas restrições impostas pelo uso único de materiais com formas pré-definidas, como tijolos e pedras. Após a queda do império romano verificou-se um largo hiato na sua utilização e durante centenas de anos o betão não foi usado, até que aproximadamente por volta dos séculos XVIII-XIX foi retomada a sua aplicação na construção civil [Shaeffer, 1992].

Num ambiente urbano torna-se difícil não nos depararmos com a sua utilização (ver figura 2.1). A sua aplicação é extremamente diversificada, sendo utilizado para construir estradas, canalizações, estruturas arquitectónicas, fundações, passeios de ruas, muros, pontes e barragens, entre muitos outros usos, sendo o material de construção feito pelo Homem mais utilizado no mundo [Lomborg, 2001].

Existem vários tipos de betão, mas na sua composição mais básica é composto por cimento, agregados finos (como as areias), agregados grossos (como as britas), água e ainda - embora opcionalmente - por adjuvantes ou adições químicas, que permitem melhorar as suas propriedades, tais como a sua trabalhabilidade e resistência, ou até mesmo dotá-lo de pigmentação. Possui ainda um outro componente,



(a) O Pantheon em Roma



(b) A Casa da Música no Porto

**Figura 2.1:** Exemplos do que o betão permitiu edificar

menos evidente: o ar - ou como é normalmente designado nesta área, o volume de vazios -, que está sempre presente e cuja quantidade deve ser cuidadosamente controlada.

## 2.2 O problema da composição do betão

Uma descrição exaustiva das características e normas regulamentares sobre o betão está fora do âmbito deste documento. Todavia, julgamos ser relevante dar uma breve introdução a este material e às suas principais propriedades e restrições a ter em conta na produção de um betão para uma obra.

Para cada obra é feito um betão à medida, normalmente numa central de betão pronto (ver figura 2.2), tendo em conta o fim a que se destina. Diversas variáveis têm de ser consideradas e de igual modo devem ser respeitadas as normas nacionais e europeias em vigor. A construção de uma barragem, de uma estrada ou de um simples muro têm, como é evidente, características perfeitamente distintas, quer a nível de complexidade, quer a nível das exigências estabelecidas pelas normas a observar.

Em seguida, são expostas de modo sucinto algumas das questões que se colocam ao realizar um estudo de composição de um betão. Destacamos os seguintes [Lourenço & Coutinho, 1986]:

- O cumprimento da classe de resistência à compressão a aplicar;
- As condicionantes ambientais da obra, tais como o tipo de clima, a agressividade química do meio - a influência que agentes agressivos possam ter sobre o betão, tais como gases, líquidos e

temperatura [Sika Portugal SA, 1999] (se estará exposto ao mar ou a agentes corrosivos como o cloro, a título de exemplo);

- Os tipos de agregados disponíveis na região;
- A consistência desejada para o betão, a qual está directamente correlacionada com a própria resistência dos elementos estruturais a betonar e também com os meios de transporte e de colocação a utilizar;
- Outras condicionantes relativas ao tipo de peças a betonar, para evitar a segregação do betão;
- A apreciação dos níveis de qualidade e a conformidade com normas europeias.

Perante estas condições, pretende-se realizar uma mistura de todos os componentes, que seja a mais compacta possível, de modo a satisfazer do melhor modo os requisitos pretendidos.

O estudo de composição implica ter um perito da área, normalmente alguém com extensa experiência de campo. Existem poucas pessoas com este perfil e habitualmente são muito procuradas, sendo que as obras estão dependentes destes peritos.



**Figura 2.2:** Uma central de betão pronto

## 2.3 O cálculo da composição do betão

### 2.3.1 A composição do betão

Tendo por referência para a composição o volume de  $1m^3$  de betão, temos que este será preenchido por todos os agregados utilizados, cimento e água, e ainda por um inevitável volume de vazios. Isto pode ser traduzido pela seguinte “fórmula fundamental da composição do betão”[Lourenço & Coutinho, 1986]:

$$c + m + A + Vv = 1000 \text{ litros } (1m^3) \text{ de betão} \quad (2.1)$$

em que

$c$  - volume absoluto do cimento em litros;

$m$  - volume absoluto dos agregados (areias, britas, ...) igual à soma dos volumes absolutos de cada uma das classes de agregados utilizadas, sendo que os volumes são medidos em litros e referidos ao volume de 1000 litros ( $1m^3$ ) de betão;

$A$  - volume da água de amassadura em litros;

$Vv$  - volume de vazios em litros; este é difícil de quantificar com precisão, pelo que são utilizados valores de referência para o efeito.

### 2.3.2 Metodologia para o cálculo da composição do betão

Em qualquer engenharia, é vital que se utilizem metodologias bem definidas para se tentar obter os melhores resultados, de um modo eficiente. Os métodos de cálculo de composição de betão não são excepção, pelo que apresentamos abaixo uma breve descrição de uma metodologia proposta para o efeito [Lourenço et al., 2004].

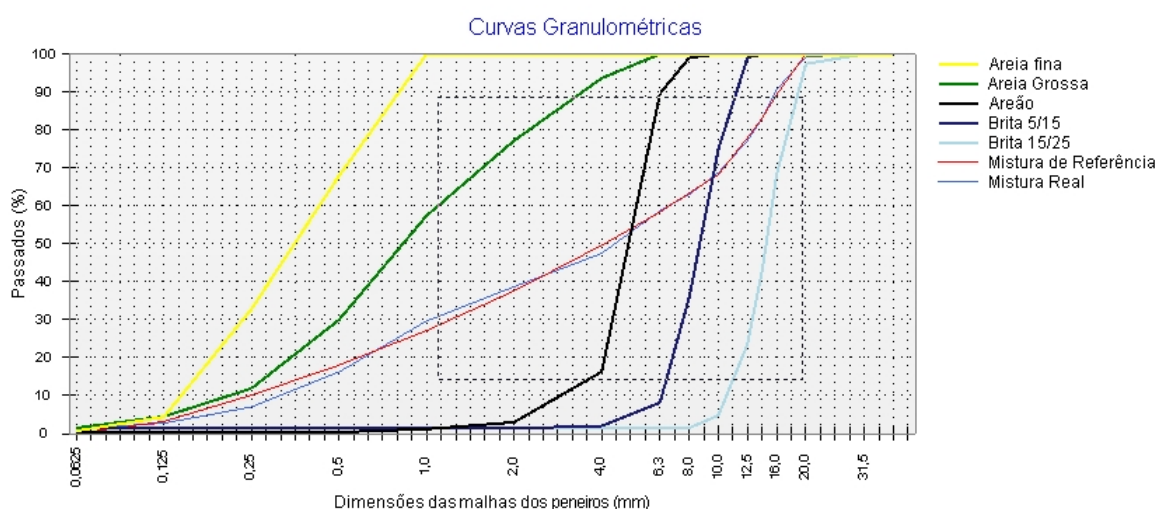
Quem deseja realizar um estudo de composição do betão, começa por delinear qual o comportamento desejado para o mesmo. Na prática, é comum considerarem-se mais factores, tais como restrições orçamentais, entre outros.

Estes dados inicialmente fornecidos podem não estar perfeitamente definidos pelo cliente, procurando-se logo aqui preencher eventuais lacunas na formulação do pedido de estudo. Assim que o comportamento do betão esteja bem estipulado, passa-se à fase da escolha e caracterização dos seus constituintes.

É então altura de definir a compacidade, tendo em conta a dimensão máxima dos agregados utilizados e da consistência do betão. Em seguida quantificam-se os constituintes da pasta ligante do betão - constituída pela parte ligante (cimento e adições) e pela parte fluída (água, adjuvantes e ar).

Sabendo-se qual o volume total destinado aos agregados, este é distribuído pelas várias classes de granulometria (uma por cada agregado), em proporções que permitam obter uma mistura com uma curva o mais próxima possível da curva granulométrica de uma proporção ideal de agregados.

Como está ilustrado na figura 2.3, cada curva do gráfico representa uma classe de agregados; estão também ilustradas as curvas que representam a mistura real e a mistura ideal dos mesmos. Neste exemplo temos portanto uma curva para a areia fina, uma para a areia grossa, uma outra para uma brita 5/15 e por aí adiante. Como regra geral, podemos afirmar que um betão terá no mínimo dois agregados e não ultrapassará os cinco, a menos que se trate de uma construção especial, como uma barragem por exemplo. Todavia essas situações excepcionais não são contempladas neste texto.

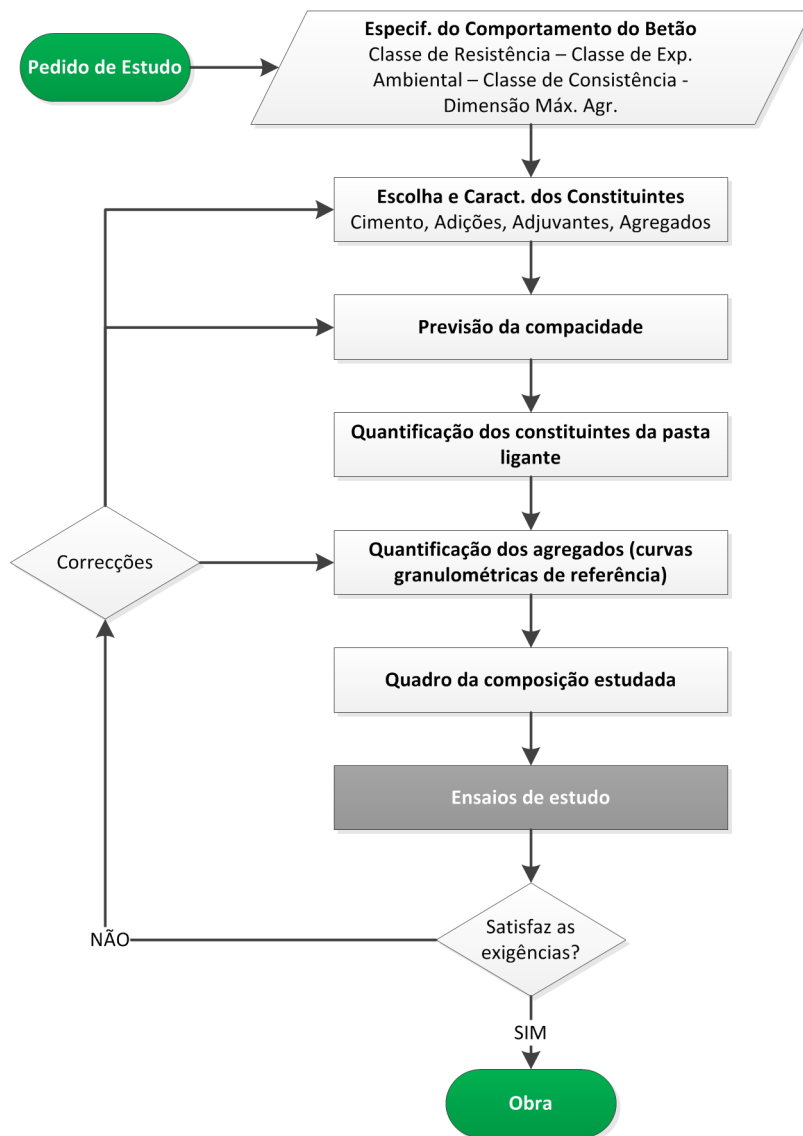


**Figura 2.3:** Curva granulométrica de referência e curva da mistura real

Existem no gráfico da figura 2.3 duas curvas especiais:

- A *curva granulométrica de referência*, que representa uma mistura granulométrica ideal, a qual confere ao betão a compacidade prevista, para as condições de colocação necessárias. É esta a mistura de agregados que serve como referência na composição do betão ideal para os nossos propósitos.
- A *curva da mistura real* representa a proporção dos agregados no estudo que estamos presente-mente a realizar. Deve estar o mais próxima possível da curva granulométrica de referência e, no limite, deveria sobrepor-se a esta. O objectivo final é que a mistura real seja o mais idêntica possível à curva ideal, sendo visível na figura 2.3 o modo como esta “serpenteia” a curva de referência.

A metodologia descrita para o estudo da composição de betões é mostrada na figura 2.4.



**Figura 2.4:** Fluxograma do processo da composição de betões de granulometria óptima[Lourenço et al., 2004]

## 2.4 Os Sistemas Periciais

Neste trabalho a aproximação escolhida para o problema em questão foi o desenvolvimento de um sistema pericial (SP). Nesse sentido é importante descrever em linhas gerais o que é um SP e em que consiste, temas que são abordados nos pontos seguintes.

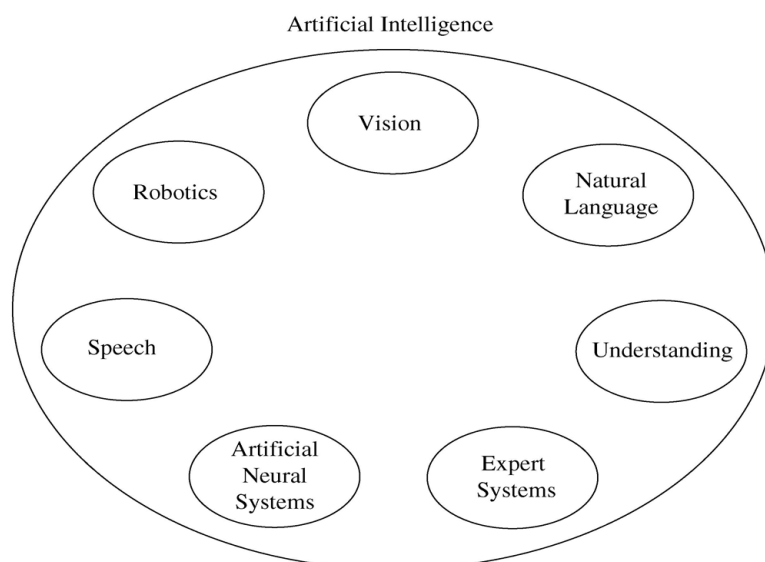
## 2.4.1 A Inteligência Artificial

Os computadores e as pessoas funcionam de modos muito distintos. Por esse motivo temos que o que para um humano é complicado e moroso, como realizar cálculos complexos, é uma das tarefas que são executadas com mais proficiência por um computador. Por outro lado, simples coisas da nossa vida diária, que até mesmo uma criança consegue fazer sem dificuldade, como reconhecer caras e objectos, são empreitadas bastante difíceis para uma máquina.

Os sistemas periciais, ou *Expert Systems* (ES), são uma aplicação de uma disciplina mais vasta, a *Inteligência Artificial* (IA). A IA diz respeito à programação de computadores para a realização de tarefas que são melhor realizadas por humanos, dado que envolvem processos mentais avançados, tais como aprendizagem perceptual, organização de memória e raciocínio crítico [Minsky, 1968].

A Inteligência Artificial é a emulação do comportamento humano, a descoberta de técnicas que nos permitem programar máquinas de modo a que estas simulem ou até aumentem as nossas capacidades mentais [Jackson, 1999].

Na figura 2.5 estão ilustradas algumas das principais aplicações da IA, onde se incluem os sistemas periciais.



**Figura 2.5:** A Inteligência Artificial e os sistemas periciais [Giarratano & Riley, 2005]

## 2.4.2 Descrição geral de um Sistema Pericial

Existem várias definições possíveis para descrever o que é um sistema pericial. Segundo [Jackson, 1999], “Um *sistema pericial* é um programa de computador que representa e raciocina com conhecimento de um assunto específico com vista a resolver problemas ou fornecer aconselhamento”. Desenvolvendo um pouco este conceito, pode-se dizer que um sistema pericial poderá realizar uma função

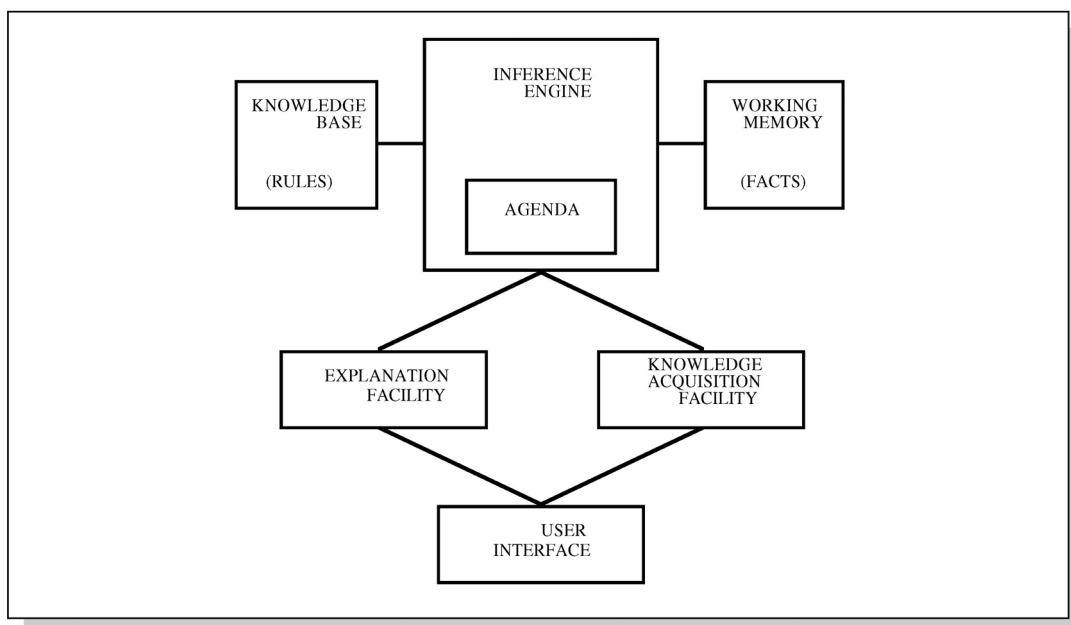
que normalmente requer um perito humano, ou poderá desempenhar o papel de assistente de um decisor humano (o utilizador). Este decisor humano poderá ser por sua vez um perito numa dada área, caso em que o SP justificará a sua existência ao melhorar a produtividade do utilizador. Em alternativa, o utilizador poderá ser alguém que idealmente atinja níveis de desempenho semelhantes aos de um perito com o auxílio do SP.

É importante referir que um perito não tem forçosamente de ser uma pessoa com elevadas qualificações, mas alguém que é altamente especializado num conjunto de competências aplicáveis a uma situação particular para um determinado fim [Jackson, 1999].

Gary Riley, um antigo membro da equipa da *National Aeronautics and Space Administration* (NASA) nos Estados Unidos, possui larga experiência em investigação, e foi enquanto trabalhava na NASA que criou o sistema de regras utilizado pelo CLIPS - uma das ferramentas de criação de sistemas periciais mais difundidas no mundo. Segundo este autor [Giarratano & Riley, 2005], um SP consiste de vários componentes, os quais são descritos nas linhas abaixo. Esta descrição aplica-se a sistemas periciais baseados em regras de produção, o caso mais comum hoje em dia:

- **Interface com o utilizador** (*User Interface*) - o mecanismo através do qual o utilizador e o sistema pericial comunicam;
- **Unidade de explicações** (*Explanation Facility*) - descreve a inferência produzida pelo sistema ao utilizador;
- **Memória de trabalho** (*Working Memory*) - uma base de dados operacional dos factos utilizados pelo motor de inferência;
- **Motor de inferência** (*Inference Engine*) - realiza inferências (ou ilações) ao decidir quais são as regras cujas condições são satisfeitas por factos ou objectos, estabelece a prioridade das regras e consequentemente dispara a regra com a maior prioridade;
- **Agenda** - uma lista de regras com a sua respectiva prioridade, criada pelo *motor de inferência*, cujos padrões são satisfeitos por factos ou objectos da *memória de trabalho*;
- **Base de Conhecimento** (*Production Memory* ou *Knowledge Base*) - local onde se encontram as regras que representam o conhecimento do sistema;
- **Unidade de aquisição de conhecimento** (*Knowledge Acquisition Facility*) - Um modo automático de adicionar conhecimento ao sistema sem a necessidade de ser expressamente introduzido no código por quem o concebeu.

O último componente, a *Unidade de aquisição de conhecimento*, é opcional em alguns sistemas. É também um componente interessante e uma mais-valia em termos de Inteligência Artificial, dado que



**Figura 2.6:** A estrutura de um sistema pericial segundo [Giarratano & Riley, 2005]

permite a dedução de regras pelo próprio sistema pericial sem intervenção humana. Porém, ao mesmo tempo que enriquece e expande o conhecimento do Sistema, também traz consigo o inconveniente de não explicar o porquê da regra que foi gerada automaticamente - o que não acontece quando um humano insere uma nova regra, introduzindo em simultâneo a sua explicação. Por este motivo é importante que o conhecimento gerado desta forma seja validado por um perito antes de ser incorporado na base de conhecimento e utilizado pelo SP.

O modo como estes diferentes componentes interagem está ilustrado na figura 2.6.

Podemos ver que através da *User Interface*, são introduzidos factos no sistema (como respostas do utilizador por exemplo), de modo directo ou opcionalmente através da *Knowledge Acquisition Facility* e em seguida são armazenados na *Working Memory*. O *Inference Engine*, o decisor do sistema, analisa os factos existentes e confronta-os com as regras existentes na *Knowledge Base*, seleccionando eventuais regras para serem colocadas na *Agenda* para posterior disparo ou execução.

Em retorno, após disparo das regras relevantes existentes na *Agenda* (a qual também controla a ordem de execução das mesmas) é apresentada ao utilizador uma decisão ou resultado acompanhado de explicações, através da *Explanation Facility*.

Num sistema de regras, o *Motor de Inferência* determina que regra será disparada, verificando quais os seus antecedentes, caso existam, que são satisfeitos pelos factos. O *Motor de Inferência* é o “cérebro” do sistema, que processa a informação constante da Base de Conhecimento para formular novas conclusões.

Ainda segundo [Giarratano & Riley, 2005], dois métodos principais de inferência são normalmente considerados: *forward chaining* e *backward chaining*.

O tipo de raciocínio seguido pelo *forward chaining* parte dos factos para as conclusões resultantes desses mesmos factos. Por exemplo, se verificarmos que está a chover antes de sair de casa (facto), então devemos levar um guarda-chuva (conclusão). A lógica do *forward chaining* faz com que seja utilizado muitas vezes em sistemas de prognóstico (por exemplo, meteorologia), monitorização e controlo (se certos factos tiverem lugar, então despoleta alertas por exemplo). Este é também talvez um tipo de raciocínio mais semelhante ao humano, que utilizamos em diversas situações diariamente.

A lógica oposta é utilizado pelo *backward chaining*. A cadeia de raciocínio parte de uma hipótese, ou seja, de uma potencial conclusão, para os factos que a suportam. Seguindo um exemplo análogo ao anterior, se não soubermos como está o tempo lá fora mas alguém entra em casa com a roupa molhada e um guarda-chuva, então é formulada a hipótese de que está a chover. Para suportar esta hipótese, podemos perguntar a esta pessoa se está a chover. Se a resposta for afirmativa, a hipótese é verificada e transformada num facto. Este método é especialmente utilizado em sistemas de diagnóstico, onde a partir de um sintoma ou observação se tenta chegar a uma causa e conseqüente acção a realizar.

Existem ainda sistemas mistos, que suportam ambos os métodos de inferência.

A programação de um sistema pericial segue uma lógica declarativa, existindo uma separação bem distinta entre o conhecimento do sistema - a Base de Conhecimento - e o Motor de Inferência, responsável por inferir novo conhecimento a partir do existente. Este é um claro contraste com a tradicional programação procedimental (como a programação em linguagem C por exemplo), em que não existe uma distinção clara entre a lógica do sistema e a parte deste que manipula os dados.

O facto de existir esta separação quase estanque facilita muito a manutenção da base de conhecimento, e reduz drasticamente a necessidade de reescrever código do sistema, tornando os sistemas periciais mais flexíveis.

### 2.4.3 Tópicos fundamentais dos Sistemas Periciais

Dada a sua natureza e idiosincrasias, existem alguns tópicos inerentes aos sistemas periciais aos quais é necessário prestar atenção para melhor percebermos a sua construção e lógica interna.

Uma descrição possível destes é apresentada, baseada na realizada por [Jackson, 1999]:

1. **A aquisição do conhecimento** - é necessário estabelecer um método para adquirir o conhecimento pericial necessário que mais tarde será traduzido para regras no sistema. O procedimento habitual é a realização de uma série de entrevistas com um ou mais peritos na área do conhecimento em causa, normalmente realizadas por um especialista informático.

Existem várias razões que tornam este processo em um dos potencialmente mais morosos e que mais trabalho exigem de todo o processo, pois é preciso que o perito não só tenha tempo disponível

- bastante tempo - mas também que tenha a sensibilidade necessária para compreender e aceitar um sistema que pode ser encarado como um seu substituto.

Por outro lado, pode ser difícil ou até mesmo penoso para alguém da área da informática entrar dentro de uma temática que não domina e que possui termos técnicos específicos e um modo de execução que desconhece inteiramente. Acima de tudo, é bastante exigente traduzir muitas das informações que o perito utiliza por vezes de um modo quase emocional, por outras palavras torna-se complicado traduzir o bom senso utilizado em muitas decisões por um especialista humano.

2. **A representação do conhecimento** - este tópico trata do modo como o conhecimento, muitas vezes de grandes quantidades de informação, é formalmente descrito de modo a ser interpretado em computação simbólica. Este terá de ser traduzido para uma linguagem ou notação com uma determinada sintaxe e semântica.

Esta representação deve ser feita de um modo adequado, para que seja possível efectuar todas as distinções necessárias na informação, contendo os detalhes e variedade exigidos - ou seja, sem prejuízo para a qualidade da informação.

Concomitantemente deve conter *poder heurístico*, ou seja, ter a capacidade de resolver problemas através da utilização da representação escolhida. Quanto mais expressiva for a linguagem, logo detentora de mais distinções semânticas, mais difícil será a inferência sobre a resolução de problemas.

Em muitos dos casos, o volume de dados a codificar é muito considerável. Por esse motivo, idealmente as expressões das regras devem ser fáceis de escrever e de fácil compreensão - sendo mesmo possível compreender o seu significado sem aprendizagem do modo como o computador as irá interpretar. Esta é uma das principais mais-valias das linguagens declarativas, conforme é descrito no final do ponto 2.4.2.

Existem diferentes métodos de representação do conhecimento. Hoje em dia o mais comum é utilização de regras, mas existem outros. Segue-se uma pequena enumeração das técnicas mais representativas [Ramos, 2009]:

- **Regras**

Conforme foi referido atrás, nos sistemas periciais actuais a maior parte do conhecimento é representado no formato de regras. Estes são muitas vezes referidos como sistemas de produção ou sistemas baseados em regras.

As regras têm o formato

*regra Identificador : Se <LHS> Então <RHS>*

onde:

- *regra Identificador* - representa o nome da regra
- *<LHS> (Left-Handed Side)* - universalmente conhecido como o lado esquerdo da regra, onde se encontram as condições que terão de ser satisfeitas para que a regra seja aplicada
- *<RHS> (Right-Handed Side)* - o lado direito da regra, as acções a realizar caso as condições no *<LHS>* se verifiquem

Um exemplo em pseudo-código pode ser visto abaixo [Giarratano & Riley, 2005]:

```
Rule: Red_light
IF
    the light is red
THEN
    stop
```

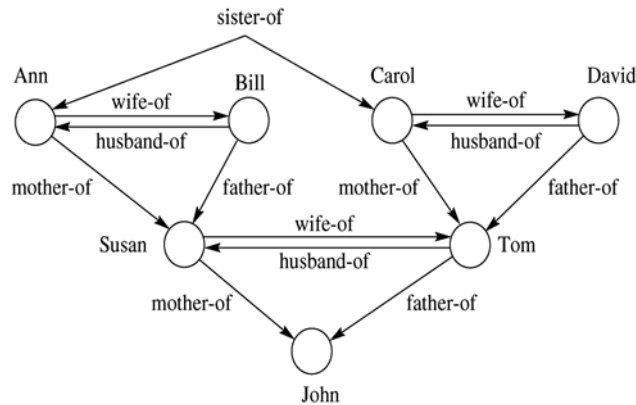
```
Rule: Green_light
IF
    the light is green
THEN
    go
```

- **Frames**

Sistema de representação criado por Marvin Minsky, um dos nomes mais importantes da Inteligência Artificial, permitem representar conhecimento acerca de um conceito ou objecto. Este paradigma é semelhante ao utilizado na programação orientada a objectos (OO), em que as propriedades (atributos) de um *frame* são representados em *slots*, aos quais por sua vez são atribuídos valores (*slot values*), como está ilustrado na figura 2.7. Estão igualmente contempladas relações de herança e a criação de instâncias dos *frames*, à semelhança do modelo OO.

FRAME			
FRAME NAME	FRAME TYPE	SLOTNAME	SLOTVALUE
			SLOTVALUE
		SLOTNAME	SLOTVALUE
			SLOTVALUE

**Figura 2.7:** A representação de um *frame* [Rattanaprteep & Chittayasothorn, 2007]



**Figura 2.8:** Uma rede semântica representada por um grafo [Giarratano & Riley, 2005]

- **Redes Semânticas**

Método de representação do conhecimento baseado num grafo em que os seus nós simbolizam objectos ou propriedades e seus valores. Os arcos do grafo representam por sua vez as relações entre os nós, frequentemente representadas como IS-A<sup>1</sup> e IS-PART<sup>2</sup> mas podendo no entanto assumir outros valores, como é possível ver na figura 2.8.

3. **O controlo do raciocínio** - é necessário um certo nível de planeamento e controlo na aplicação prática do conhecimento existente no sistema, para saber quando colocar questões (ou saber o que perguntar) e que validações realizar, entre outros. Para além da própria arquitectura do sistema, existem diferentes mecanismos que permitem controlar a execução de regras, por exemplo, estabelecendo prioridades ou fazendo a separação por grupos de regras com a mesma afinidade. Trata-se acima de tudo de uma questão de metodologia.
4. **A explicação das soluções** - este módulo ajuda o utilizador a entender melhor o porquê de algumas decisões do sistema, através de explicações. É bastante importante pois transmite maior confiança ao utilizador, ao compreender melhor o processo e percebendo o porquê dos resultados, em vez de simplesmente ser confrontado com uma deliberação do software. Torna-se mesmo mais compreensível para o próprio perito, aquando do teste do sistema, entender a linha lógica condutora utilizada pelo SP - deste modo poderá validá-lo mais facilmente.

Em conclusão, resta dizer que, num cenário ideal, um sistema pericial poderia ser capaz de igualar ou até mesmo superar o perito. Se considerarmos que um sistema pode incluir o conhecimento de vários peritos, pelo menos teoricamente é possível que este seja, em pelo menos algumas situações, superior ao

<sup>1</sup>uma relação de classificação, pode ser interpretada como *é do tipo*

<sup>2</sup>uma relação de pertença, pode ser interpretada como *é parte de*

perito humano, dado que terá mais informação complementar e diferentes métodos para a resolução de uma dada questão, logo mais “rico”.

Por outro lado, podemos também considerar que a informação disponibilizada por mais de um perito pode mesmo ser contraditória, dado que podem eventualmente utilizar métodos que, não obstante serem eficazes, têm abordagens ao problema inteiramente distintas. Isto pode tornar a tradução e implementação deste conhecimento complicada, e aí reside a grande importância dos tópicos atrás referidos, como a representação do conhecimento e o controlo do raciocínio.

O perito tem algo que é difícil de traduzir em código, a capacidade de discernimento. Ao ver um resultado que não faz sentido, é capaz de perceber que algo está errado. Em contrapartida, as pessoas são afectadas pelo cansaço e pelo stress e podem tomar decisões erradas.

Não é suposto um sistema pericial ser infalível, tal como as pessoas não o são.

Na prática julgamos não ser errado afirmar que o sistema pericial será um sucesso se conseguir igualar ou aproximar-se do desempenho de um perito humano.

## Capítulo 3

# Estado da Arte

*“Emprega o teu tempo a melhorar-te pelos escritos de outros homens, para ganhares facilmente o que outros trabalharam arduamente para atingir.”*

— Sócrates

Neste capítulo é feita uma síntese dos principais trabalhos que foram realizados até hoje no campo dos estudos de composição de betão, com especial ênfase para os sistemas periciais. Para uma introdução mais completa, e dado que este foi o tipo de sistema desenvolvido nesta dissertação, a primeira secção é dedicada às ferramentas para desenvolvimento de sistemas periciais.

### 3.1 Ferramentas para desenvolvimento de Sistemas Periciais

Para a concepção de um sistema pericial é necessário dispormos de ferramentas que os permitam criar e desenvolver. Existem várias terminologias para descrever este tipo de software, por vezes encontramos a designação ferramenta de desenvolvimento, outras vezes *expert system shell* ou apenas *shell*, plataforma, *framework* ou simplesmente linguagem.

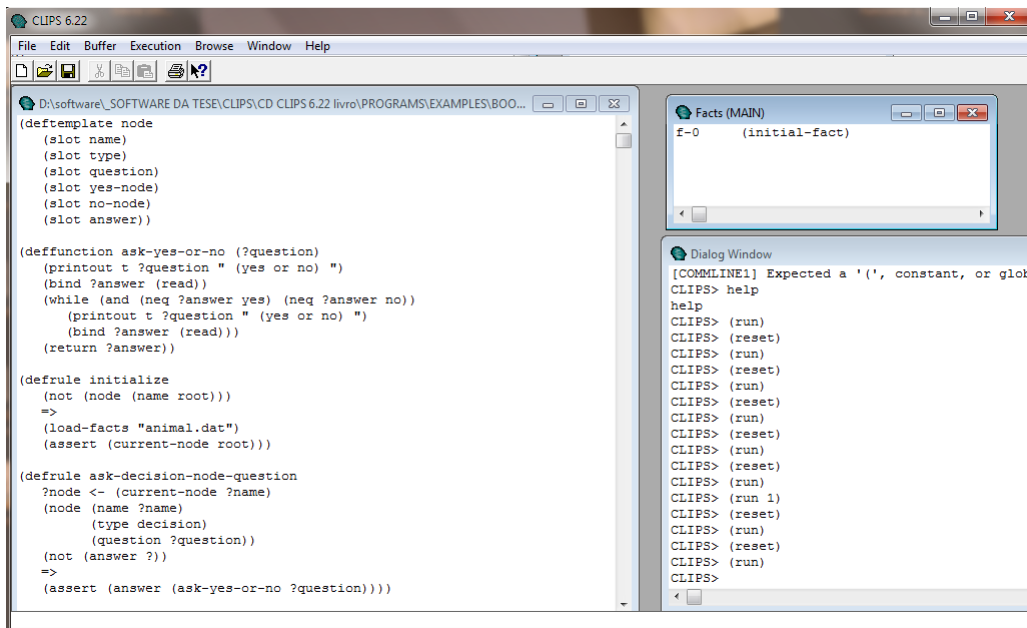
Das várias ferramentas disponíveis, e tendo em mente que existem outras não abrangidas nesta recolha, tentamos descrever e analisar algumas das mais populares, apresentando uma selecção:

- **CLIPS**

O *C Language Integrated Production System* (CLIPS) é uma linguagem criada por uma equipa da NASA no *Johnson Space Center*, da qual fazia parte Gary Riley, para o desenvolvimento de sistemas periciais com a capacidade de inferência e representação de conhecimento do OPS5<sup>1</sup>, embora

---

<sup>1</sup>ver descrição desta linguagem na página 25



**Figura 3.1:** O ambiente de programação do CLIPS 6.2 no S.O. *Windows 7*

mais potentes. O seu mecanismo de inferência é o *forward chaining*, e suporta três diferentes paradigmas de programação: regras de produção, programação orientada a objectos e programação procedimental (tal como a linguagem *C*, na qual foi escrito) [Giarratano & Riley, 2005]. Hoje em dia é largamente utilizado em diferentes áreas, a nível académico, governamental e na indústria. O CLIPS pode correr nos sistemas operativos *Windows*, *MacOS* e *Unix*. No entanto, a portabilidade é possível para qualquer sistema que possua um compilador suportado para *ANSI C* ou *C++*. A versão standard do CLIPS oferece um ambiente de programação interativo (ver figura 3.1), incluindo ferramentas para *debugging*, ajuda *on-line* e um editor. Possui uma documentação completa e inclui funcionalidades para verificação e validação de sistemas periciais [Riley, 2012].

- **Prolog**

Muitos sistemas periciais foram construídos nesta linguagem e três características muito propiciaram este facto: programação baseada em regras, *pattern matching* nativo e execução *backtracking*. O facto de ser baseada em regras permite uma programação essencialmente declarativa. O *pattern matching* e o mecanismo de *backtracking* permitem a inferência e o controlo do raciocínio, essenciais em qualquer sistema pericial. As regras em *Prolog* são utilizadas para a representação do conhecimento e o seu motor de inferência para gerar conclusões [Merritt, 1989]. Na figura 3.2 é possível ver um dos ambientes de desenvolvimento mais populares para *Prolog*, o *LPA Prolog*, assim como código desta linguagem.

Figura 3.2: LPA Win-Prolog v4.920 [Logic Programming Associates, 2012]

- **Drools / JBoss Rules**

O Drools é uma *Business Logic integration Platform* (BLiP), escrita em Java. É um projecto *open source* com o apoio da JBoss e Red Hat. Existem várias versões, na altura da escrita deste texto a mais actual é a versão 5.4.0.

A primeira versão (1.0) foi lançada em 2001 e começou com a utilização de uma pesquisa por força bruta. Foi reescrito na versão 2.0, desta vez já baseado no algoritmo *Rete*<sup>2</sup>, o que melhorou em muito o seu desempenho. As regras eram principalmente escritas em XML.

A versão 3.0 introduziu o formato *drl*, uma linguagem especificamente concebida para a escrita de regras e sofreu uma série de melhorias no código. A versão 4.0 trouxe ainda mais desenvolvimentos e o primeiro lançamento do *Business Rules Management System* (BRMS). Este formou a base da nova versão, a 5.0, onde o Drools se tornou uma *Business Logic integration Platform*, a qual consiste dos seguintes módulos [Bali, 2009]:

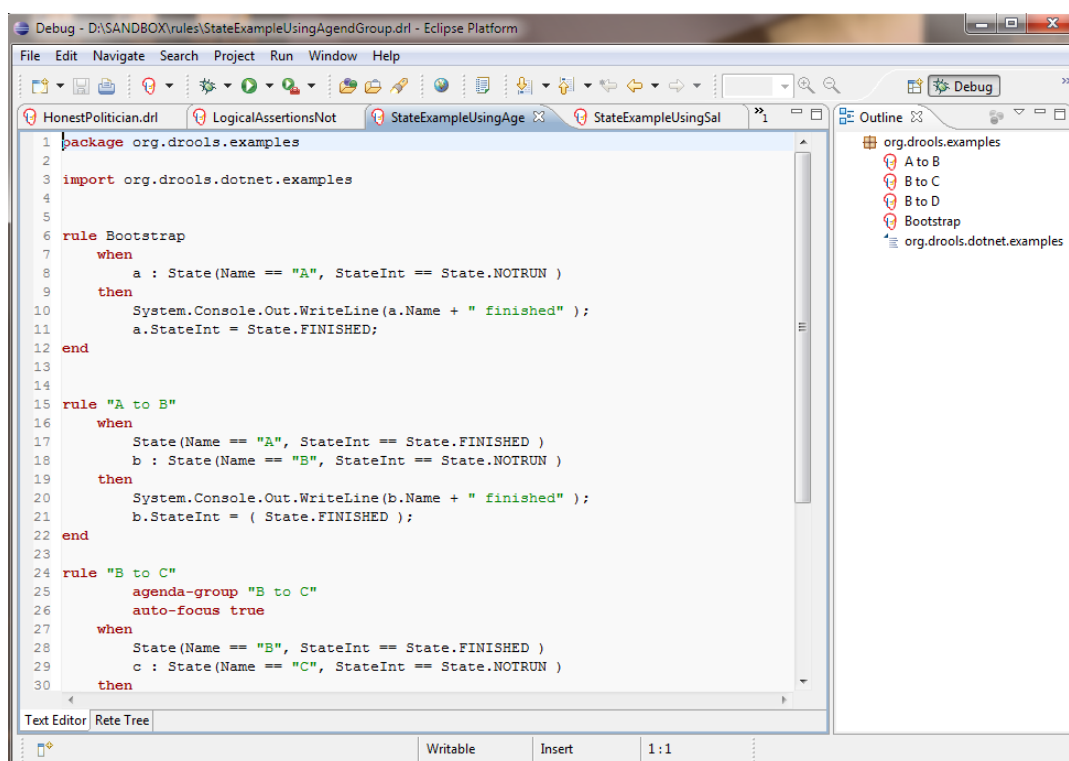
1. *Drools Expert* - o motor de inferência das regras;
2. *Drools Fusion* - módulo *Complex Event Processing* (CEP);

---

<sup>2</sup>um método muito eficiente para comparação de uma colecção grande de padrões (*patterns*) com uma colecção grande de objectos, sendo muito utilizado em sistemas periciais baseados em regras [Albert, 1989]. A sua descrição será desenvolvida mais à frente no capítulo 5.

3. *Drools Flow* - o *workflow* combina regras e processos;
4. *Drools Guvnor* - Business Rules Management System;
5. *Drools Solver* - trata-se de módulo opcional, um algoritmo de pesquisa construído com base no motor de inferência do Drools para a resolução de problemas de planejamento (como a criação de horários).

Para o seu desenvolvimento, existe um *plugin* para o Eclipse IDE, com suporte para escrita e debug de código e dos processos, o qual pode ser visualizado na figura 3.3.



**Figura 3.3:** O Eclipse IDE com o plugin de suporte ao Drools

- **Jess**

Jess, ou *Java Expert System Shell*, é uma implementação do CLIPS para Java com novas funcionalidades. Trata-se de um sistema de produção, com uma linguagem declarativa baseada em regras, para a criação de sistemas periciais. Possui uma sintaxe semelhante ao LISP<sup>3</sup>, em que cada componente (factos, regras, funções, etc) tomam a forma de uma lista. Abaixo é possível ver um exemplo de uma regra em Jess [Strauss, 2007]:

<sup>3</sup>LISP, ou *LISt Processing*, é uma das mais antigas linguagens de programação de alto-nível e uma referência em IA

```

(defrule ask::ask-question-by-id
"Given the identifier of a question, ask it and
assert the answer"
(declare (auto-focus TRUE))
(MAIN::question (ident ?id) (text ?text) (type
?type))
(not (MAIN::answer (ident ?id)))
?ask <- (MAIN::ask ?id)
=>
(bind ?answer (ask-user ?text ?type))
(assert (answer (ident ?id) (text ?answer)))
(retract ?ask)
(return))

```

É importante também referir outras ferramentas de desenvolvimento, que não obstante a maior parte já se encontrar desactualizada, tiveram um papel relevante no desenvolvimento de sistemas periciais anteriores, especialmente nas décadas de 80 e 90. Destacamos em seguida algumas das principais, utilizadas pelos SP que são revistos mais à frente, no ponto 3.2.1:

- **Personal Consultant**

Trata-se de um produto da Texas Instruments, implementado em IQLISP (uma versão de LISP). Utiliza regras e *frames*<sup>4</sup> e suporta *forward* e *backward chaining*. É utilizada em computadores e também na TI LISP. A interface com o utilizador é baseada em janelas e tem um módulo de explicações [Cohn & Harris, 1990].

- **EXSYS**

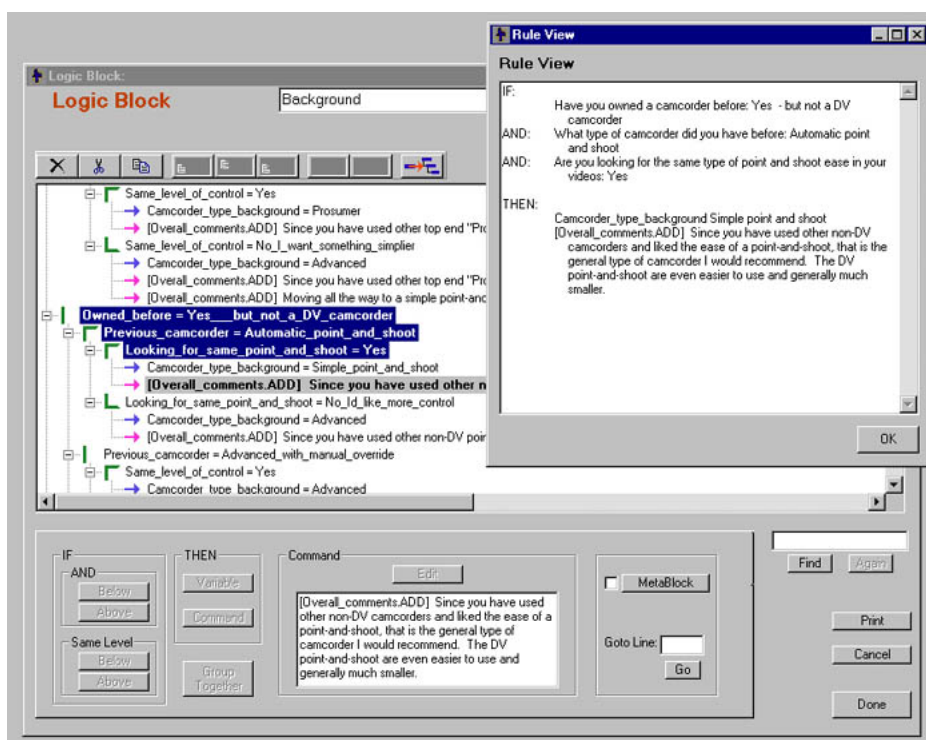
Nos anos 80, este era um produto da empresa EXSYS Inc., implementado na linguagem de programação C, utilizando regras do tipo IF-THEN-ELSE. Este sistema utilizava *backward chaining* que incluía um módulo *runtime* e um gerador de relatórios. Podia servir de interface a outros produtos para fornecer *frames* e função de *blackboard*<sup>5</sup>. Podiam ser utilizadas chamadas a programas externos para aquisição de dados e execução de rotinas [Cohn & Harris, 1990].

A Exsys acompanhou o evoluir dos tempos e ainda hoje tem uma série de produtos para criação de sistemas periciais e outros relacionados, dos quais se destaca o *Exsys Corvid Expert System*

---

<sup>4</sup>uma aplicação semelhante à da programação orientada a objectos para sistemas periciais concebida por [Minsky, 1974]; cada *frame* é semelhante a um objecto, por sua vez composto por *slots*, representando os atributos do *frame*

<sup>5</sup>paradigma para a resolução de problemas descrito por [Corkill, 1991], em que a questão a resolver é especificada e em seguida diferentes fontes adicionam alternadamente conhecimento de forma a se complementarem com o objectivo de se chegar a uma solução válida



**Figura 3.4:** O Exsys Corvid Expert System Development Tool da EXSYS

*Development Tool* para a criação de sistemas periciais online (ver figura 3.4). Desde 2000 fez uma aposta forte na vertente *web* dos seus produtos e actualmente há uma série de empresas de renome que utilizam sistemas periciais construídos com os seus produtos, em diferentes áreas de negócio e aplicações, dos quais se destacam apenas alguns [EXSYS, 2012]:

- *Jet Aircraft Diagnostics and Customer Support* (Cessna Aircraft Corporation)
- *Restoration of Shipboard Power Systems During Battle* (US Navy)
- *Auto Industry Improved Operations through Complex Diagnostics* (Ford Motor Company)
- *Tax and Legislative Auditing and Reporting* (PricewaterhouseCoopers)
- *Online Computer Systems Configuration and Report Generation* (Hewlett-Packard)
- *Real-Time Manufacturing Process Control* (Nestlé)

- **LEVEL5 (anteriormente Insight2+)**

Produzido pela Level Five Research, é implementado em *PASCAL* e disponível para computadores e também para máquinas VAX. O sistema é baseado em regras e suporta *backward* e *forward*

*chaining*. Os factos inseridos são representados como objectos com atributos. Pode aceder a programas externos e a bases de dados e a sua interface é orientada a menus [Cohn & Harris, 1990].

- **OPS5 (Official Production System, Version 5)**

Linguagem de desenvolvimento com origem na Carnegie Mellon University, disponível na Digital Equipment Corporation (DEC) para uso em equipamentos VAX. É a base do R1/XCON (Expert Configurer), um sistema pericial para configuração de sistemas VAX e a primeira aplicação comercial de um sistema pericial. Utiliza regras de produção e o mecanismo de *forward chaining*, tendo sido desenvolvida no início dos anos 80 por Charles L. Forgy [Forgy, 1981], o criador do algoritmo *Rete*. Foi a primeira linguagem com regras de produção a ser utilizada num sistema pericial [Brick, 2006]. A OPS5 e as suas linguagens derivadas têm sido utilizadas desde então e muitos SP modernos ainda a utilizam hoje em dia. Foi a predecessora da OPS83, a qual foi lançada comercialmente e que possuía construtores semelhantes aos da linguagem C.

- **Kappa-PC**

Introduzido em 1990 pela IntelliCorp, tratava-se de uma plataforma de desenvolvimento para sistemas periciais para ambiente Microsoft Windows, fruto da investigação em Inteligência Artificial desta empresa. Utilizava ferramentas orientadas a objectos, implementando regras para tomada de decisão também ligadas a objectos. Mais tarde foi abandonada pela IntelliCorp.

- **RuleMaster**

Foi desenvolvido pela Radian Corporation e é constituído por uma série de programas interrelacionados e escritos na linguagem C. Induz regras através da generalização de exemplos contendo atributos com valores e acções. O RuleMaster permite dois modos distintos de desenvolver as regras no sistema [Wicklunf & Roth, 1987]: através de programação na *Radial language* (semelhante a *PASCAL*) ou então de um processo de indução de regras (*RuleMaker*) que gera módulos *Radial* através de exemplos. O módulo contém uma árvore de decisão e realiza uma série de testes até atingir uma decisão que despoleta uma acção. Após a acção ter sido realizada, o controlo é passado para um novo estado no módulo. Utiliza o mecanismo de *forward-chaining*.

## 3.2 Soluções existentes para calcular a composição do betão

Conforme foi referido no Capítulo 2, um estudo de composição implica a supervisão de um perito da área, alguém com extensa experiência no terreno. Dado que existem poucas pessoas com este perfil, são normalmente muito requisitadas, o que por vezes pode ser um problema dado que a sua disponibilidade

é limitada. Por outro lado, a dispersão de locais de obra obriga a deslocações que consomem recursos financeiros e tempo.

No passado foram realizadas diferentes aproximações no sentido de encontrar alternativas ao perito, de reduzir o tempo investido por este na composição de betões, ou até mesmo para funcionarem como instrumentos pedagógicos. Algumas das mais relevantes são a seguir descritas.

### **3.2.1 Sistemas Periciais para a definição da composição do betão**

#### **3.2.1.1 BETVAL**

O BETVAL é um SP baseado em regras que fornece conselhos para a selecção de betão pronto (betão produzido em centrais de betão, por oposição ao produzido na obra). O objectivo deste sistema pericial é providenciar assistência ao pessoal presente na obra na escolha do tipo de betão fresco a encomendar à central de betão pronto.

A sua metodologia consiste em realizar recomendações baseadas em três áreas de conhecimento:

1. A classe de resistência e técnica de feitura do betão (por exemplo, a cura, o aquecimento e o tratamento por calor);
2. A classe de consistência do betão, baseado no tipo de estrutura e equipamento de produção;
3. Sugestões na escolha do tamanho máximo do agregado.

Este SP foi um de vários produzidos no Technical Research Center of Finland (VTT), cujo site pode ser consultado em <http://www.vtt.fi>. Foi desenvolvido utilizando Insight2+ (uma versão anterior do LEVEL5) para utilização num PC comum. Este sistema é um protótipo de demonstração, que foi utilizado principalmente como instrumento de aprendizagem, dado que foi admitido que necessitaria de uma expansão da sua base de conhecimento para entrar em produção [Seren, 1988; Kaetzel & Clifton, 1991].

#### **3.2.1.2 COMIX**

O COMIX é um sistema baseado em conhecimento que utiliza regras e *frames*, realizando recomendações para a mistura de composição de betão. Foi desenhado para ser utilizado por técnicos de betão, engenheiros e consultores da área.

Este SP define a composição do betão baseado nas normas para construção de betão da Nova Zelândia, mais especificamente no *New Zealand Code Specification for Concrete Construction*. O COMIX calcula a quantidade de cimento, agregados grossos e agregados finos para 1m<sup>3</sup> de betão. Relaciona o tipo de estrutura a construir com a consistência necessária para o betão, assim como o modo de colocação do mesmo. O sistema recomenda a proporção de água/cimento necessária a partir de uma determinada

resistência desejada e calcula a quantidade de cimento. O volume de agregados é então calculado e a massa dos constituintes da mistura do betão é também calculada e mostrada ao utilizador.

O SP foi desenvolvido pelos *Central Laboratories* na Nova Zelândia e o conhecimento constante do sistema é proveniente de peritos locais certificados [Smith, 1987; Kaetzel & Clifton, 1991].

### 3.2.1.3 CONCEX (CONCrete EXpert)

Este sistema pericial baseado em conhecimento foi desenhado para fornecer assistência na verificação da qualidade do betão na obra. O sistema procura oferecer uma explicação dos factores que afectam a qualidade do betão, testa-o a diferentes estágios de maturação para prever a qualidade e disponibiliza ao utilizador um método de consulta e explicação do processo lógico utilizado.

A base de conhecimento do CONCEX representa informação obtida de livros, publicações, manuais e peritos na área. O SP é constituído por 5 módulos:

1. cálculo da resistência do betão;
2. composição do betão e suas propriedades;
3. diagnóstico da consistência do betão ou do conteúdo de ar do mesmo;
4. previsão da resistência à compressão;
5. previsão da resistência à compressão a diferentes estágios.

O CONCEX foi desenvolvido pela Rutgers University, utilizando a ferramenta de desenvolvimento *RuleMaster*. O conhecimento é representado na forma de regras IF-THEN e inclui programas escritos em *C* e *FORTRAN*, os quais realizam os cálculos necessários. O sistema foi concebido para ser utilizado num IBM PC compatível [Williams et al., 1991; Kaetzel & Clifton, 1991].

### 3.2.1.4 Concrete Mix Designer

O Concrete Mix Designer é um sistema pericial baseado em regras que calcula as proporções dos constituintes do betão para a realização de ensaios, de acordo com o método ACI (método para composição de betão do *American Concrete Institute*). Foi desenvolvido com a ferramenta *Personal Consultant Plus*, uma shell de desenvolvimento da Texas Instruments, no *Department of Civil and Architectural Engineering of the University of Miami* e foi pensado como uma ferramenta para estudantes de engenharia e engenheiros civis.

Integra rotinas criadas em *BASIC* para fazer perguntas e respostas na interacção com o utilizador. O sistema justifica as suas conclusões e pode expandir a sua base de conhecimento, a qual é de fácil

compreensão. Este conhecimento está representado por regras do tipo IF-THEN, agrupadas por sua vez em *frames*. Cada *frame* representa um componente do betão, tal como um agregado grosso ou uma areia.

Só é aplicável a betão de peso regular com resistência à compressão entre 17 a 35 MPa e não contempla o uso de adições ou adjuvantes na mistura [Malasri & Maldonado, 1988a; Abdullahi et al., 2008] e [Malasri & Maldonado, 1988b; Williams et al., 1991].

### **3.2.1.5 EXMIX**

O EXMIX é um sistema baseado em regras e orientado a objectivos para realizar a composição de misturas de betão, definindo as proporções necessárias dos seus constituintes. Possui dados qualitativos e quantitativos na sua base de conhecimento e utiliza a *shell* para sistemas periciais LEVEL5. Selecciona proporções para a água de mistura, cimento e agregados, considerando também adições indutoras de ar (especialmente úteis em países frios, para proteger o betão dos ciclos de gelo/degelo) e a humidade existente nos agregados. Foi desenvolvido de acordo com a norma CSA A23.1-M90 (Canadian Standards Association 1990) e a ACI 211.1-89 utilizando o método com massa ou com volume, estando também conforme com a norma ACI 318-89, respeitante à qualidade exigida do betão [Akhras & Foo, 1994; Abdullahi et al., 2008].

### **3.2.1.6 Knowledge-Based Expert System for Concrete Mix Design**

Também baseado em regras, este SP foi concebido para ajudar o utilizador a fazer a composição de um betão, assim como ajustes. O sistema considera os seguintes critérios relativos ao betão: trabalhabilidade, consistência, resistência, durabilidade, densidade, adições/adjuvantes, transporte e temperatura do ar. Suporta a composição de diferentes tipos de betão, tais como betão leve, betão normal, betão pesado e incorpora um factor de confiança na escala de 0 a 10 para representar o nível de certeza no resultado. Utiliza a *shell* EXSYS, baseada em regras, e exporta a composição numa folha de cálculo QUATTRO PRO. O conhecimento constante da sua base é proveniente de várias origens, desde manuais do *American Concrete Institute* (ACI), normas ASTM (*American Society for Testing and Materials*) e entrevistas e levantamentos junto de peritos [Bay & Amirkhanian, 1994; Abdullahi et al., 2008].

### **3.2.1.7 HPCMIX**

O HPCMIX é um protótipo desenvolvido por Zain et al. da *Universiti Kebangsaan Malaysia* para o sistema operativo Microsoft Windows, o qual estabelece a proporção necessária para ensaios de misturas de *High Performance Concrete* (HPC) e recomendações necessárias para ajustes. O seu conhecimento provém de várias origens, textuais (desde manuais técnicos a artigos científicos) e de peritos na feitura de betão, embora a componente mais forte seja a literatura. O método utilizado para gerar a composição do betão foi o desenvolvido por [Aïtcin, 1998], dada a sua grande aceitação na Malásia. Este tem também a

limitação de apenas poder ser aplicado a betões cuja resistência à compressão está entre 40 a 160 MPa. O processo começa por seleccionar cinco características ou proporções de materiais na seguinte sequência:

1. relação água/ligante;
2. conteúdo de água;
3. dosagem de superplastificante;
4. quantidade de agregados grossos;
5. volume de vazios.

É possível ver o fluxograma desenvolvido para o conhecimento adquirido [Zain et al., 2005], relativo ao ajuste da mistura, na figura 3.5. São considerados três critérios para avaliar o desempenho de uma mistura (resistência, trabalhabilidade e durabilidade). No caso de uma composição falhar e com base nos testes realizados, são exploradas as diferentes possibilidades de modo a encontrar a causa dessa mesma falha.

O sistema foi desenvolvido utilizando a *shell* para sistemas periciais Kappa-PC com uma técnica híbrida de representação do conhecimento, ou seja, utiliza regras e *frames*. Por outras palavras, o sistema foi modelado com uma abordagem orientada a objectos e regras de produção. A interacção entre objectos, instruções e processos foram codificadas na forma de *métodos*, *funções* e *regras*, os quais partilham uma sintaxe comum. Os métodos foram escritos em *Kappa-PC Application Language* (KAL) e guardados nos objectos. As funções também são escritas em KAL, existindo previamente uma biblioteca com mais de 300 funções nativas. Abaixo é possível ver um exemplo de uma regra simples em pseudocódigo e em KAL [Zain et al., 2005]:

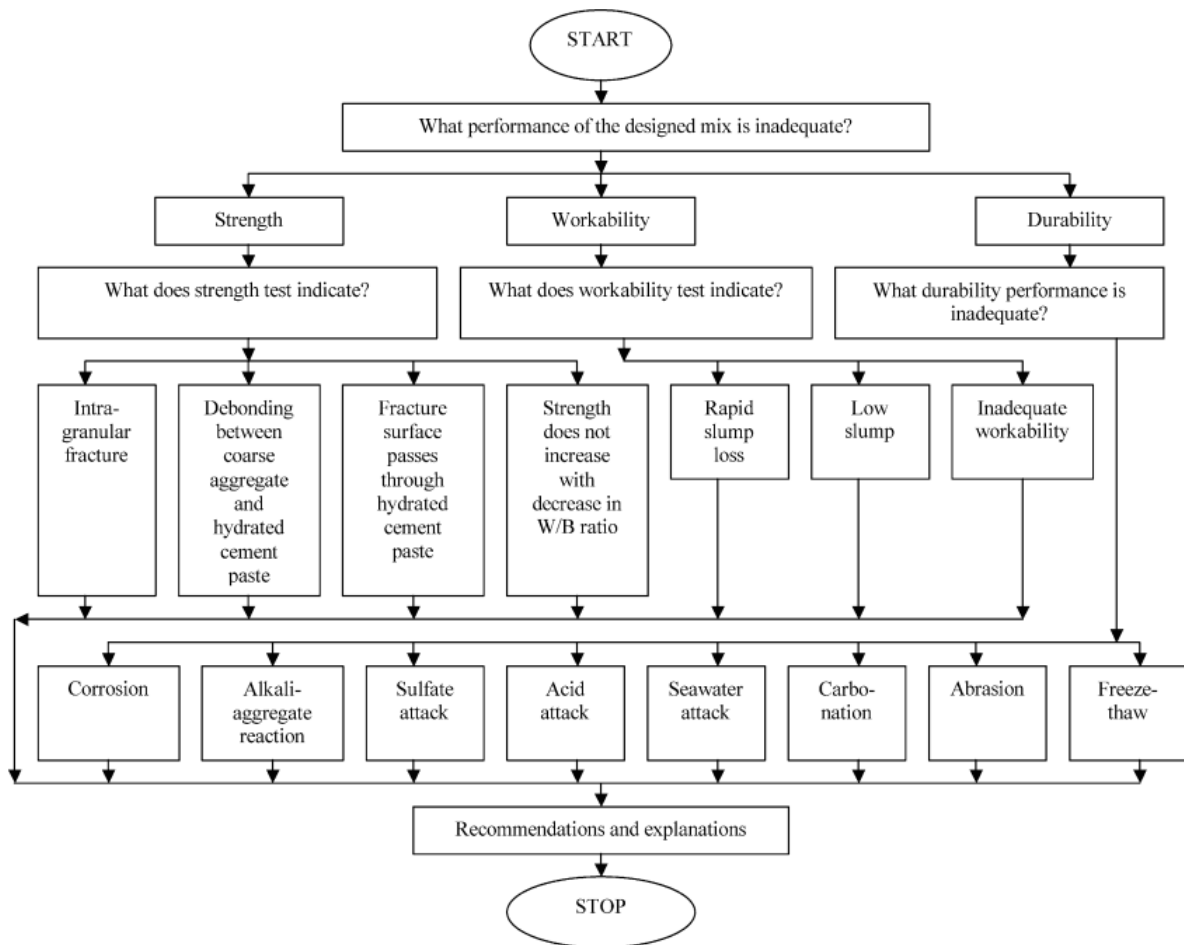
```
If: Shape of the coarse aggregate is cubic
Then: Coarse aggregate content of the mix should be 1100 kg/m3.
```

*Em KAL, esta regra seria escrita assim:*

```
If: CoarseAgg: ShapeOfAgg #ZCubic
Then: CoarseAgg: ContentKgPerM3 #Z1100;
```

A base de conhecimento consiste de três módulos principais e um complementar. Estes são:

1. *Mix Design module* - criação das proporções dos constituintes do betão (ver figura 3.6);
2. *Mix Performance Adjustment module* - ajuste das proporções após ensaios de laboratório (ver figura 3.7);



**Figura 3.5:** Fluxograma de ajuste da mistura no HPCMIX

3. *Cost Estimation module* - estimativa aproximada por metro cúbico de betão;
4. *General Information module* - presta assistência complementar ao utilizador na utilização do HPCMIX.

Em suma, é capaz de seleccionar proporções para a água de mistura, cimento, adições, agregados e superplastificante. Considera ainda os efeitos do ar presente no betão e no superplastificante e a humidade nos agregados. Como é habitual em vários sistemas periciais, possui módulo de explicações, o qual pode ser incrementalmente aumentado, e a sua base de conhecimento é de fácil compreensão. Pode também ser utilizado como uma ferramenta educativa [Zain et al., 2005].

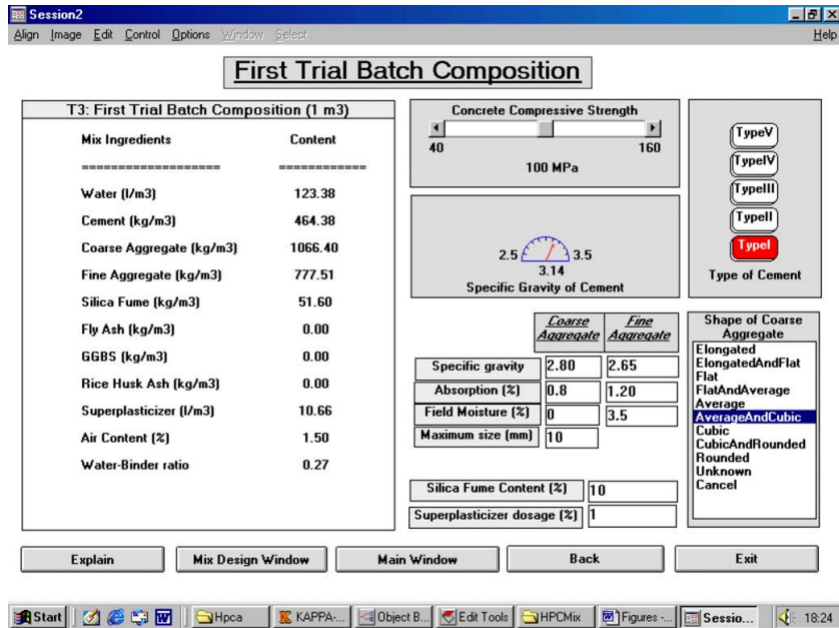


Figura 3.6: Composição do betão no HPCMIX para o primeiro ensaio

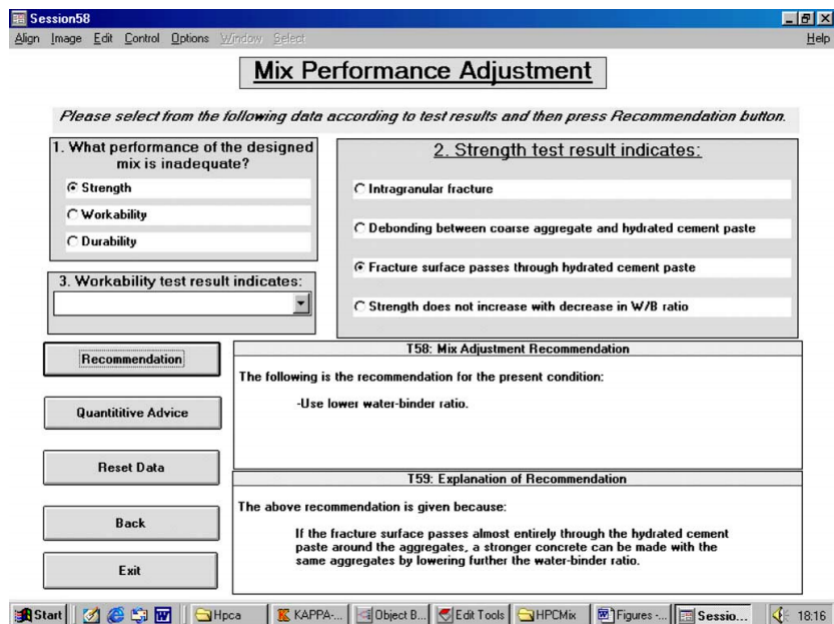


Figura 3.7: Janela do módulo de *Mix Performance Adjustment* do HPCMIX

### 3.2.1.8 The application of automatic acquisition of knowledge to mix design of concrete

Este sistema pericial utiliza um paradigma bastante diferente dos anteriormente apresentados, pois trata-se de um sistema de aquisição automática de conhecimento baseado em redes neuronais. O objectivo, no entanto, é o mesmo: criar uma mistura o mais ideal possível para um betão. Segundo os seus autores [Ji-Zong et al., 1999], esta aproximação pode resolver alguns dos problemas enfrentados pelos profissionais da área, pois embora conheçam boas e comprovadas misturas para betão, quando confrontados com um novo projecto têm quase sempre de começar do zero.

Esta abordagem baseada em redes neuronais segue os três passos a seguir descritos:

1. construir sistemas de bases de dados através da recolha de dados de misturas bem sucedidas;
2. definir vários modelos de redes neuronais para extrair e armazenar as regras de proporções para misturas de betão, através da aprendizagem de exemplos da base de dados;
3. criar misturas de betão através da implementação do sistema de aquisição de conhecimento; podem ser visualizadas misturas no ecrã e também simular o efeito de alterações nas misturas (pelas alterações nas propriedades do betão como a resistência e trabalhabilidade).

O sistema foi programado em *Visual C++* (portanto para ambiente *Windows*) e é composto por três modelos, sendo que cada um é constituído por uma base de dados para composição da mistura, base de conhecimento, um bloco de aprendizagem para redes neuronais e um bloco de solução de problemas. Estes modelos são:

- *Mix-design model* (o módulo principal);
- *Slump-prediction model*;
- *Strength-prediction model*.

O *Mix-design model* recebe do utilizador os *inputs* classe de cimento, dimensão máxima dos agregados, tipo de agregado grosso, módulo de finura<sup>6</sup>, consistência (ou *slump*), resistência à compressão a 28 dias e tipo de adição ou adjuvante. Gera então um vector de *outputs* de materiais como o cimento, água, agregados finos e grossos, adição e/ou adjuvante e cinzas volantes<sup>7</sup>. O vector de *inputs* juntamente com o vector de *outputs* constituem um par de vectores de aprendizagem. Dúzias de pares destes vectores de aprendizagem por sua vez constituem um conjunto de amostra de aprendizagem.

A informação útil para a composição do betão é extraída das amostras de aprendizagem por uma rede do tipo *back-propagation* (BP), uma das arquitecturas para redes neuronais mais comumente utilizadas

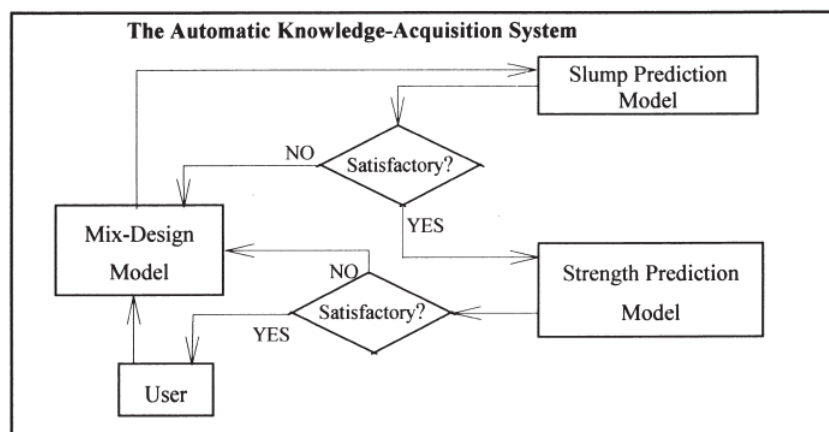
---

<sup>6</sup>factor calculado a partir da análise granulométrica, que serve como indicador do comportamento de uma mistura de betão com agregados de uma certa granulometria

<sup>7</sup>um tipo de adição para melhorar certas características do betão

devido em parte à sua capacidade de aprendizagem de mapeamentos multidimensionais [Hecht-Nielsen, 1989]. Quando um determinado nível de aprendizagem é atingido, a rede BP pode ser considerada na base de conhecimento para a composição da mistura do betão e está pronta a ser utilizada.

O *Slump-prediction model* e o *Strength-prediction model* são utilizados para assegurar que a composição do betão gerada pela rede neuronal é satisfatória. Em caso negativo, o utilizador pode ajustar alguns parâmetros de modo a obter outra mistura e realizar novamente o teste até obter o resultado desejado. Se o sistema for incapaz de obter uma solução que satisfaça os requisitos pretendidos, o utilizador pode realizar ele próprio uma composição de betão, a qual pode ser posteriormente adicionada à base de conhecimento do SP. Com isto pretende-se que esta seja continuamente melhorada [Ji-Zong et al., 1999]. A figura 3.8 ilustra uma visão de alto-nível do processo geral.



**Figura 3.8:** A arquitectura do sistema de auto-aquisição de conhecimento de [Ji-Zong et al., 1999]

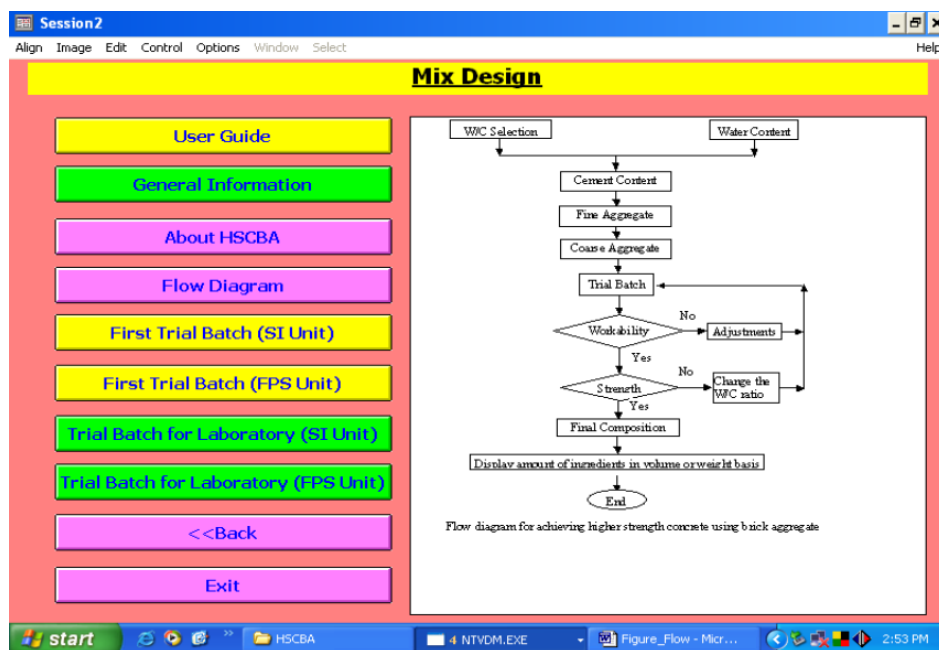
### 3.2.1.9 ESBAC - An Expert System for Mix Design of Brick Aggregate Concrete

Este sistema pericial é um protótipo para composição de um tipo específico de betão, em que os agregados são compostos por tijolo reciclado, em vez das tradicionais areias ou britas. Segundo os autores [Salam et al., 2012], isto é em parte devido ao facto de este SP ter sido desenvolvido no Bangladesh, onde embora a procura pelos agregados grossos seja alta, os depósitos de rocha natural são escassos. Todavia, o tijolo é aqui uma fonte alternativa de agregados e também uma opção mais ecológica e económica. Em contrapartida, a criação da mistura cria outros desafios devido à maior capacidade de absorção e menor peso dos tijolos em relação aos agregados comuns.

O ESBAC segue as normas americanas e britânicas para a composição de betão e foi concebido para betões com resistência à compressão de até 28MPa, sendo capaz de seleccionar as proporções de cimento, água de mistura e agregados necessárias.

O seu modo de operação é o seguinte: o utilizador começa por entrar no módulo de *Mix Design* (ver figura 3.9) e em seguida inicia o “First Trial batch”, onde terá de introduzir os vários *inputs* necessários (ver figura 3.10). O módulo de explicações pode ser activado neste momento para qualquer ajuda necessária. Assim que o SP detém todos os dados necessários, é possível ver o resultado do ensaio, como está patente na figura 3.11.

O conhecimento presente na sua base foi retirado da literatura técnica, assim como por entrevistas com 4 engenheiros civis com larga experiência em composição de betão com agregados de tijolo, estando representado através de regras de produção e de *frames*, sendo portanto um sistema híbrido. A *shell* para sistemas periciais Kappa-PC foi utilizada para o desenvolvimento, pois possui características que se ajustam a este tipo de representação híbrida do conhecimento [Salam et al., 2012]. As regras são escritas em KAL, sendo muito semelhantes às da listagem do ponto 3.2.1.7. O SP possui igualmente um módulo de explicações, pensado para estudantes de engenharia civil, como é possível ver na figura 3.12.



**Figura 3.9:** Janela do *Mix Design* do ESBAC mostrando o fluxograma do mesmo



Figura 3.10: Os inputs necessários para o primeiro ensaio de laboratório



Figura 3.11: Janela do ESBAC com a composição do betão e dados do primeiro ensaio

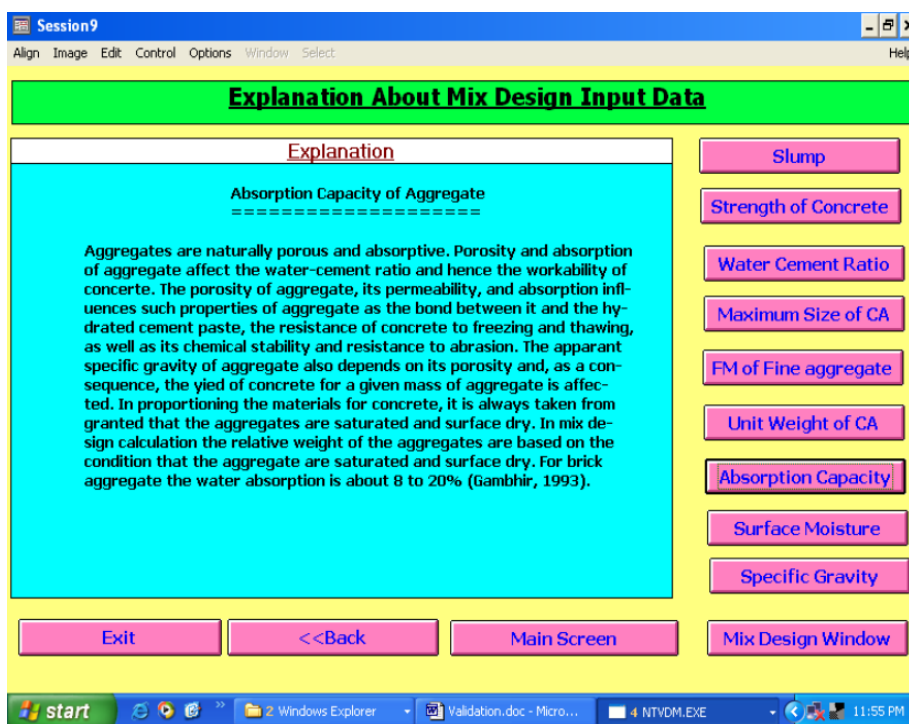


Figura 3.12: Uma explicação do ESBAC durante a entrada de dados

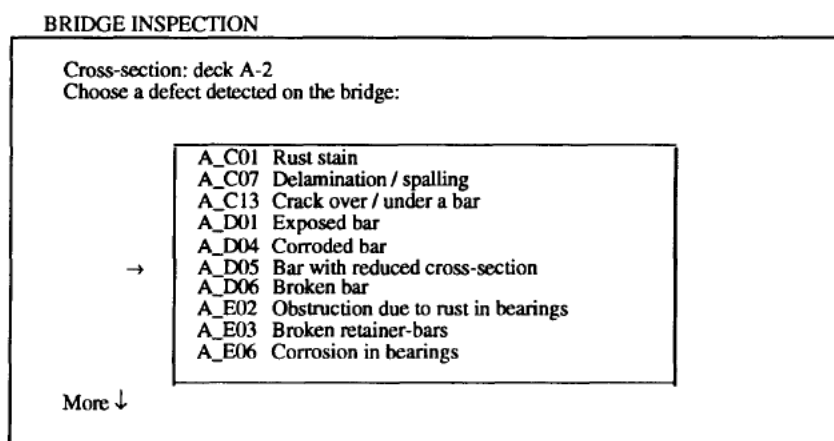
### 3.2.2 Outros Sistemas Periciais para betão

Os sistemas periciais atrás apresentados estão directamente ligados à problemática da composição do betão. Todavia, existem outros estudos sobre sistemas periciais para betão com outras aplicações, como o diagnóstico de falhas no betão, ou então para diferentes tipos deste material, como o betão de alto desempenho ou resistente a terramotos.

Os sistemas apresentados neste ponto, embora não sejam dedicados claramente à composição de betão, representam outros trabalhos relacionados na área do betão, servindo como um complemento ao estado da arte.

#### 3.2.2.1 Expert System for Concrete Bridge Management

Os autores deste sistema pericial, J. de Brito et al., explicam que existe há vários anos um dilema entre construir uma nova ponte ou repará-la, por questões orçamentais e dadas as altas taxas de deterioração que são observadas. Neste sentido, em 1994 foi criado um protótipo de um SP para a gestão de pontes de betão, cuja funcionalidade foi dividida em 2 módulos [Brito et al., 1997]:



**Figura 3.13:** Lista de defeitos do BRIDGE-1

- **BRIDGE-1** - um sistema interactivo baseado em conhecimento;
- **BRIDGE-2** - módulo de optimização de estratégias de gestão, o qual por sua vez é composto por três sub-módulos, descritos mais à frente.

O protótipo considera apenas defeitos relacionados com a corrosão e problemas relacionados com a deterioração dos materiais. Os seus principais propósitos são auxiliar nas inspecções e providenciar conhecimento pericial na tomada de decisões.

O primeiro módulo, BRIDGE-1, é utilizado num computador portátil no local da ponte para inspecções periódicas, providenciando informação útil relativa aos defeitos detectados e preparando os relatórios da inspecção. De modo a normalizar os procedimentos, foi criado um sistema de classificação de defeitos passíveis de serem encontrados, agrupados em 9 conjuntos. Quando um defeito é detectado, o inspector da obra selecciona-o no ecrã (ver figura 3.13) e tem acesso às seguintes opções:

1. Método de Diagnóstico;
2. Causa Provável (ver figura 3.14);
3. Defeitos Associados;
4. Técnicas de Reparação Associadas.

O BRIDGE-2 por sua vez é para utilização no escritório e faz o tratamento de toda a informação das inspecções. Este segundo módulo avalia a evolução da segurança da ponte, define a política de manutenção e as estratégias a aplicar para os principais trabalhos de reparação da estrutura - tendo em conta uma análise económica e de fiabilidade. Conforme foi dito atrás, este módulo é constituído por três sub-módulos:

**BRIDGE INSPECTION**

DEFECT: A\_D05 Bar with reduced cross-section  
 CROSS-SECTION: deck A-2  
 TYPE OF INSPECTION: Current Inspection  
 Please select the probable Causes of the defect:

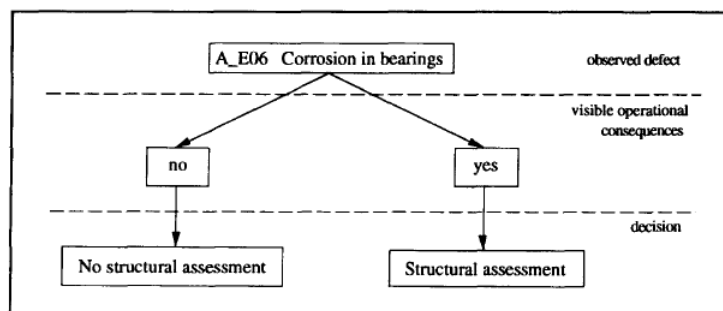
C_A14	Insufficient reinforcement / prestressing design cover
C_A24	Drainage directly over concrete, joint, bearing or anchorage
C_B09	Deficient concrete compaction / curing
C_B11	Inaccurate reinforcement / prestressing positioning / detailing
C_F01	Water (wet / dry cycles)
C_F02	Natural carbon dioxide
C_F03	Salt / salty water (chlorides)
C_G01	Water (man-caused)
C_G02	Man-caused carbon dioxide
C_G03	Man-caused deicing salts
C_A20	Excessive exposed areas in structural elements / faulty geometry
C_A23	No prevision of a minimum inclination in quasi-horizontal surfaces
C_A25	Other drainage design faults
C_A26	Lack of waterproofing membrane
C_A28	Incomplete / contradictory / over compact drawings
C_B01	Wrong interpretation of the drawings

More ↓

**Figura 3.14:** Causas sugeridas de um certo defeito (BRIDGE-1)

1. Estratégia de inspeção;
2. Manutenção;
3. Reparação.

Após cada inspeção é realizada uma análise de fiabilidade, indicando ao utilizador se uma avaliação estrutural deve ser realizada, no caso de ser detectado um defeito grave na ponte. O sistema contém regras na sua base de conhecimento para decidir se esta avaliação deve ser realizada, das quais é possível ver um exemplo na figura 3.15. Em seguida, o módulo de reparação (B2R) é utilizado para fazer a análise económica e de fiabilidade para decidir qual o tipo e a duração dos trabalhos de reparação a realizar.



**Figura 3.15:** Exemplo de uma regra da base de conhecimento do BRIDGE relativa a corrosão das estruturas

Neste protótipo, apenas os defeitos relacionados com a corrosão em pontes de betão são contemplados no sistema pericial. Estavam previstas melhorias na análise económica do módulo BRIDGE-2 para expandir o seu âmbito para além das pontes de betão.

### 3.2.3 Sistemas de outra natureza utilizados para cálculo da composição do betão

Paralelamente aos sistemas periciais, existem outras aplicações de diferentes naturezas que tentam responder às necessidades dos profissionais de engenharia civil. Neste ponto são apresentados alguns destes métodos, sempre com o objectivo de definir a composição do betão, sendo que à excepção da solução baseada em *Matlab*, reflectem especialmente a realidade nacional.

#### 3.2.3.1 Folhas de cálculo

Embora à primeira vista possa parecer deslocado, a realidade é que folhas de cálculo como o *Microsoft Excel* são das ferramentas mais utilizadas pelos profissionais responsáveis pela definição da composição do betão, numa grande parte das obras nacionais e internacionais. Habitualmente são utilizadas *worksheets* elaboradas, muitas vezes criadas pelo próprio profissional em causa, onde são definidas várias fórmulas e calculadas as quantidades necessárias de cada constituinte do betão. É comum que tenham gráficos associados, com a percentagem de cada constituinte visível, de modo a facilitar a sua compreensão.

#### 3.2.3.2 A Matlab Program for Diagnosis and Adjustment of Mix Proportions of Structural Lightweight Concrete

Este programa foi desenvolvido em *Matlab*, um popular ambiente de programação para desenvolvimento de algoritmos, análise e visualização de dados e computação numérica [Mathworks, 2012]. Funciona como um assistente para o ajuste de proporções de misturas de betão leve, e pode diagnosticar causas de insucesso no desempenho de misturas deste tipo específico de betão, fornecendo recomendações para os acertos necessários. O software proporciona uma interface gráfica ao utilizador em ambiente *Matlab*, a qual podemos ver na figura 3.16. No seu ecrã principal, está dividida em cinco partes [Abdullahi et al., 2009]:

1. Do lado esquerdo, existe um grupo com 5 menus *popup* que permitem que o utilizador seleccione o tipo de falhas de desempenho e as suas causas;
2. *Explanatory Notes*, onde são mostradas possíveis explicações para as falhas;
3. *Recommendation*, onde o utilizador pode visualizar recomendações para melhorar a mistura do betão;
4. *User's Guide*, que dá acesso a um módulo com procedimentos detalhados para realizar ajustes à composição do betão, baseado no método utilizado (peso ou volume), visível na figura 3.17;
5. *Reset*, um botão para colocar o sistema no estado *default*, limpando todos os campos.

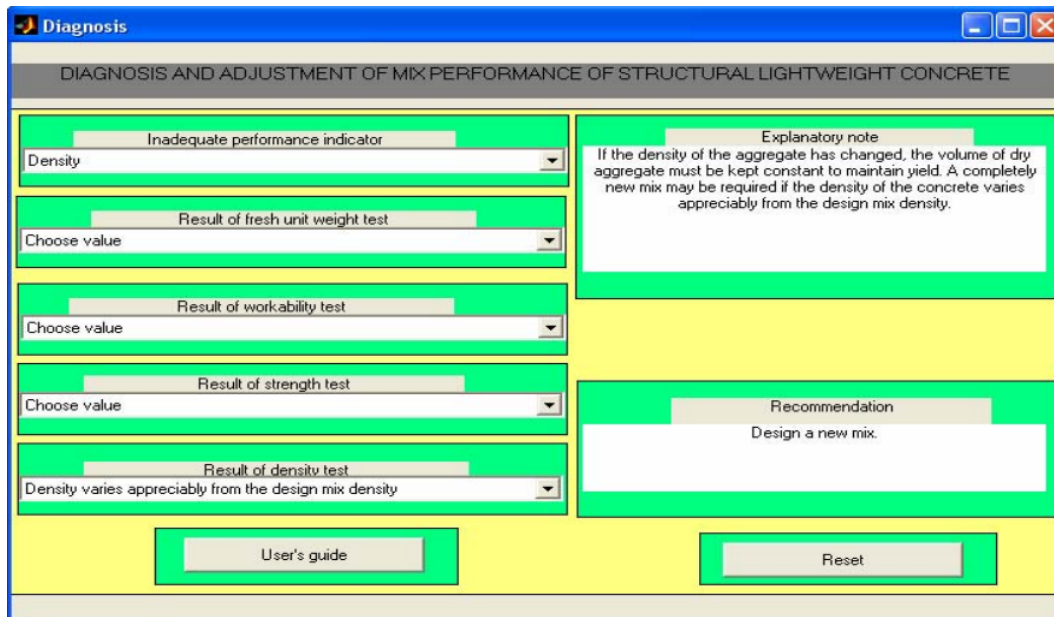


Figura 3.16: Diagnóstico, explicação e recomendações na interface do *Matlab*

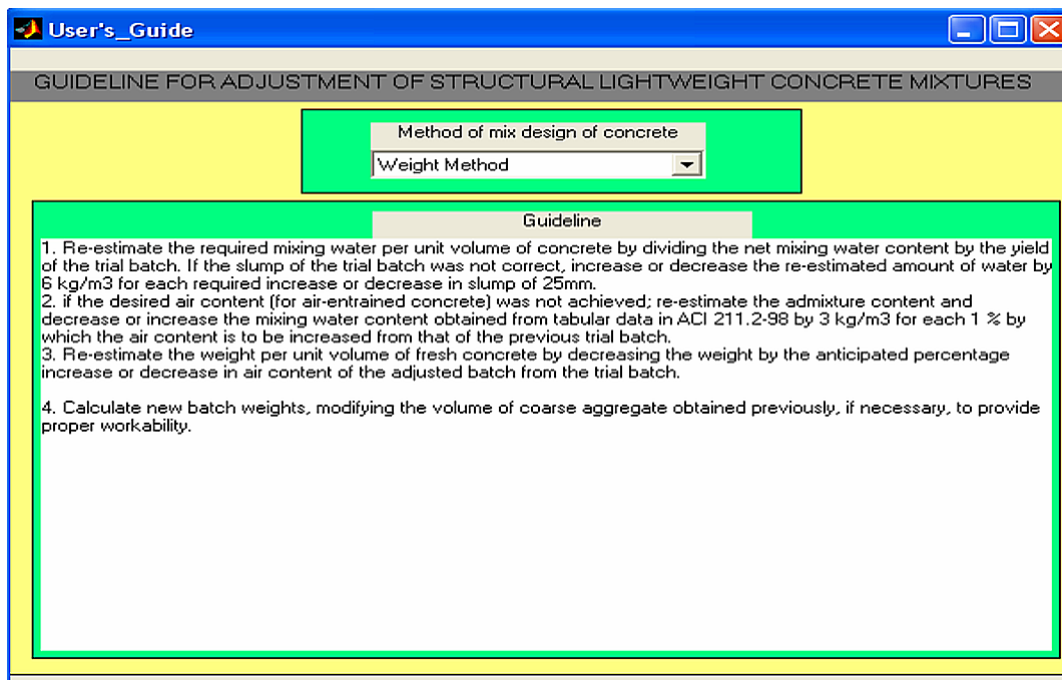
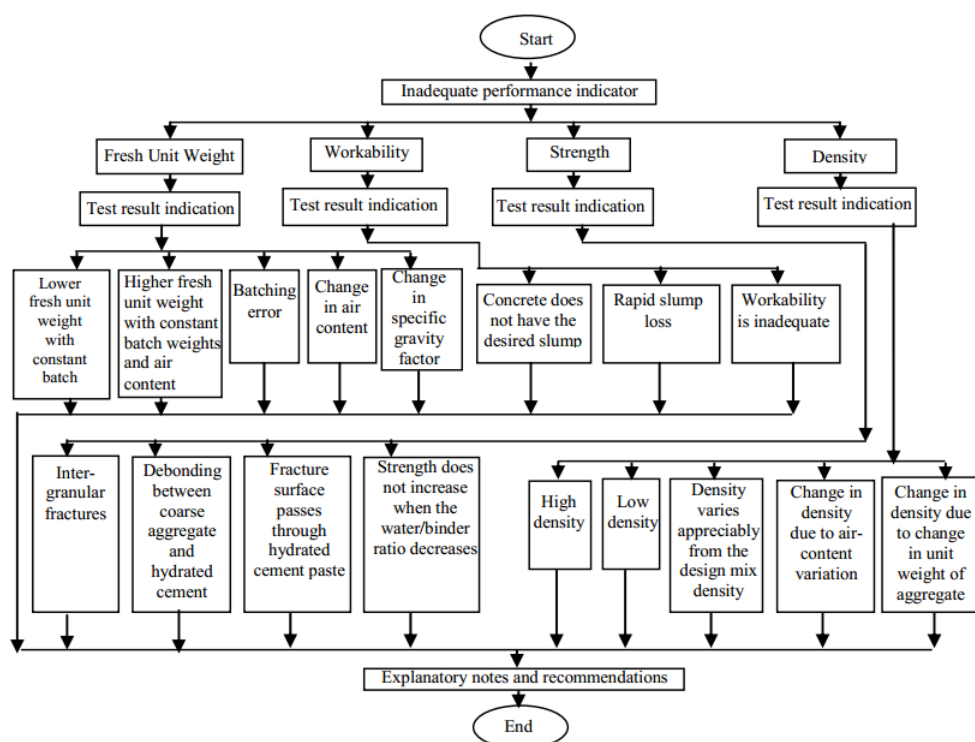


Figura 3.17: O *User's Guide* no ambiente *Matlab*



**Figura 3.18:** Fluxograma do diagnóstico para ajustes na mistura do betão, implementado em *Matlab*

A lógica inerente ao sistema de diagnóstico está patente no fluxograma da figura 3.18 [Abdullahi et al., 2009]. O sistema começa com uma rotina principal, que consiste em identificar, através de um indicador, a causa do desempenho insuficiente do betão. Para executar esta tarefa, é subdividida em subrotinas menores, e é definida uma indicação para o resultado do teste. Por sua vez, estas subrotinas são novamente partidas em outras ainda mais modulares, tais como “Workability is inadequate”, “Change in air content” ou “Low density”. Todas estas subrotinas possuem ainda outros submódulos associados com as notas necessárias para providenciar explicações ao utilizador, assim como as recomendações necessárias aos ajustes para a mistura do betão.

Este sistema retirou o seu conhecimento da literatura técnica, e não utiliza qualquer tipo de mecanismo de inferência comercial, como é habitual em sistemas baseados em conhecimento. O conhecimento é escrito como regras em *M-files*<sup>8</sup>, através de declarações IF-ELSEIF. O programa pode ser incrementalmente expandido e mantido, sem afectar o seu desempenho geral. O propósito principal deste sistema é servir como um guia e ferramenta pedagógica, especialmente a novatos na composição e ajuste de betões leves.

<sup>8</sup>ficheiros em Matlab que funcionam como *scripts*

### 3.2.3.3 MatWin - Gestão de Ensaios de Materiais

O programa MatWin é um software comercial baseado em sistema de bases de dados relacionais (Microsoft Access), construído especificamente para ensaios de materiais, ensaios sobre betões e os seus constituintes, podendo eventualmente ser extensivo a outros materiais. Pode ser fornecido em versões executáveis de Access 97 ou 2000, implicando a instalação prévia da aplicação Microsoft Access 97 ou 2000.

Este sistema de bases de dados é uma evolução dos anteriores programas BECOMP e NAB (Núcleo de Aglomerantes e Betões), desenvolvidos no período de 1987 a 1992. De acordo com o LNEC [Laboratório Nacional de Engenharia Civil, 2012], esta nova versão tem maior facilidade na sua utilização, maior qualidade dos relatórios produzidos, e pela primeira vez é possível a utilização em rede do software. Inclui já algumas análises, nomeadamente a análise dos resultados de famílias de betão e controlo da sua produção.

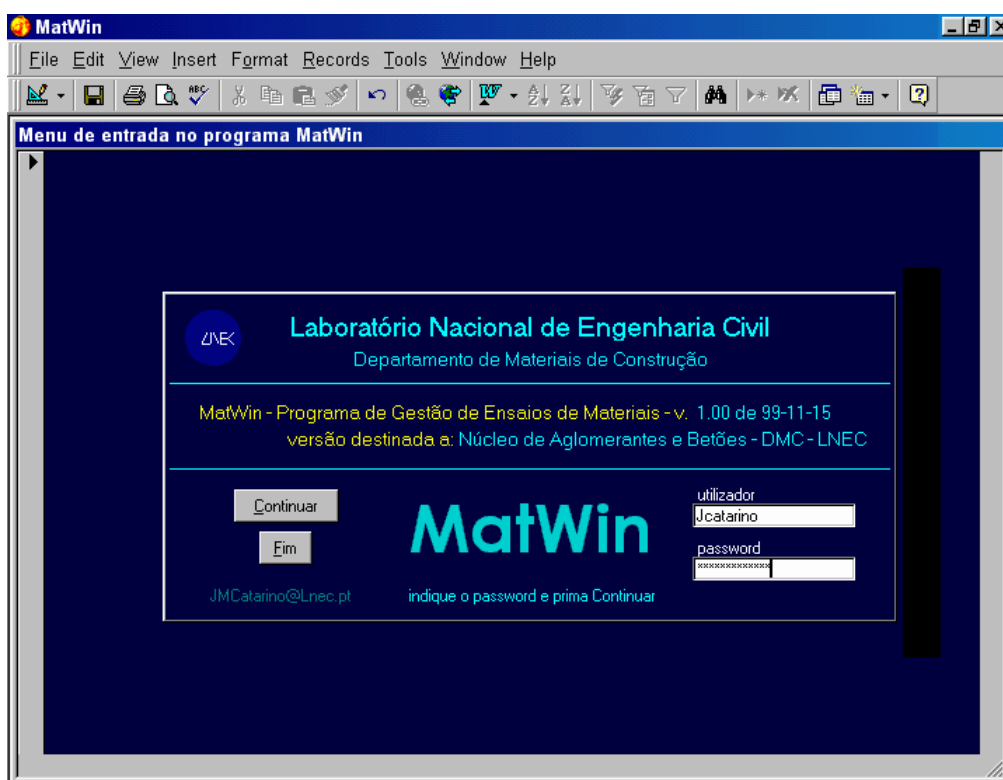
As funções principais do MatWin são as a seguir descritas [Laboratório Nacional de Engenharia Civil, 2012]:

- registo de amostras a ensaiar e de estudos de composição de betão;
- programação de ensaios e o registo dos resultados dos ensaios;
- tratamento dos resultados dos ensaios;
- cálculo da composição de betões, com optimização da sua granulometria pelo ajuste a diversos tipos de curvas de referência;
- elaboração de relatórios predefinidos, sobre conjuntos de ensaios e amostras, incluindo relatórios de análise da informação contida na base de dados;
- controlo da produção e da conformidade de famílias de betão.

Este software tem um sistema de licenciamento em que o número de utilizadores de uma empresa é ilimitado. No entanto deverá ser fornecida ao LNEC uma lista de todos os utilizadores da empresa, com os respectivos tipos e passwords. O sistema contempla os seguintes tipos de utilizador:

- Tipo 1 – administrador da base de dados (acesso a todas as aplicações);
- Tipo 2 – responsável por uma aplicação (acesso a uma aplicação);
- Tipo 3 – técnico de laboratório (acesso a uma aplicação com parte das opções);
- Tipo 4 – técnico de laboratório (acesso a todas as aplicações com parte das opções).

Na figura 3.19 podemos ver uma captura de ecrã deste software, onde é possível perceber que foi concebido para o sistema operativo Microsoft Windows 95/98.



**Figura 3.19:** Ecrã inicial da aplicação MatWin, produto do LNEC

### 3.2.3.4 SIKACOMP

O software Sikacomp 2.5 foi desenvolvido em 2007 para a SIKA Portugal S.A., a filial portuguesa da SIKA, uma multinacional suíça da indústria química para a construção. Este aplicação foi concebida para Microsoft Windows XP/Vista no Microsoft Visual Studio 2003, na linguagem *Visual Basic.NET* (VB.NET).

Este software teve dois antecessores, o Sikacomp 1.0 - a versão original, ainda para MS-DOS e desenvolvida no início dos anos 90 por Jorge Madureira - e o Sikacomp 2.0, a primeira concebida para sistemas Microsoft Windows, desenvolvida pelo mesmo autor da versão 2.5 e também desta dissertação, André David. A orientação técnica da perspectiva da engenharia civil foi da responsabilidade de Jorge Lourenço, Professor Adjunto do Instituto Superior de Engenharia de Coimbra (ISEC) e co-orientador da presente tese. A versão 2.5 representou uma maturação da 2.0 e introduziu pela primeira vez a conformidade com as normas europeias, actualmente obrigatórias por lei.

O programa permite calcular a composição de um betão e gera gráficos das granulometrias óptimas e reais, possuindo ainda um módulo de controlo de custos e geração de relatórios [Sika Portugal, 2007]. É presentemente utilizado em diversas centrais de betão pronto em Portugal e em Angola, sendo

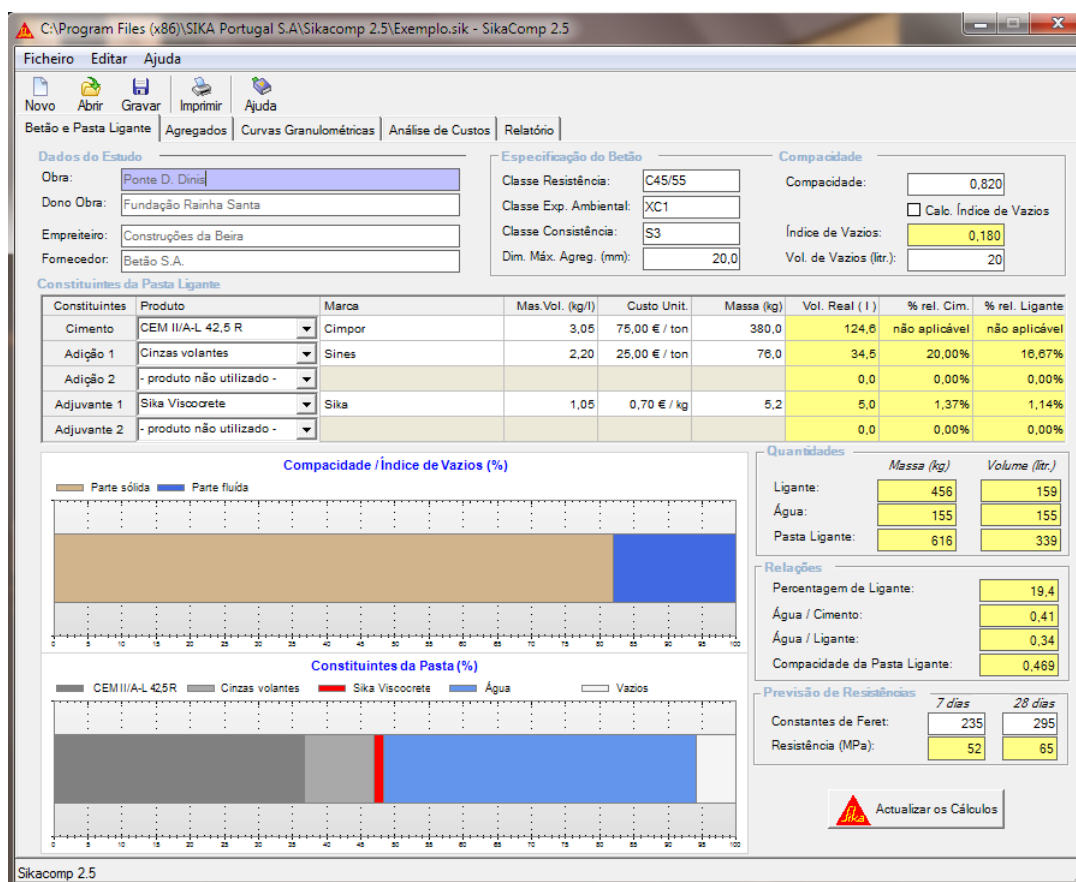


Figura 3.20: Ecrã inicial do Sikacomp 2.5

provavelmente a mais actual das aplicações deste género em uso no nosso país (ver figura 3.20).

## Capítulo 4

# Definição do Problema

*“There’s a way to do it better - find it.”*

—Thomas A. Edison

### 4.1 Identificação do problema

A opinião consensual dos profissionais da área da engenharia civil consultados para este trabalho, é a de que não conhecem um software que trate da questão da composição do betão de um modo satisfatório. Mais ainda, as poucas aplicações disponíveis, especialmente em Portugal, são consideradas antigas ou desajustadas das necessidades reais. Existe de certo modo, à altura da escrita desta dissertação, um vazio no que toca às ferramentas de trabalho destes profissionais, os quais na sua esmagadora maioria se limitam a utilizar elaboradas folhas de cálculo como o único software para o cálculo da composição do Betão.

### 4.2 Análise comparativa dos sistemas analisados

Após a apresentação dos sistemas descritos no capítulo 3, foi realizada uma comparação dos diferentes sistemas para estudos de composição de betão, a qual pode ser consultada na tabela 4.1. As características seleccionadas para serem examinadas foram as que mais divergiam nos sistemas e que simultaneamente eram mais relevantes. Passam em seguida a ser explicadas:

- *Ferramenta de desenvolvimento* - qual a ferramenta (linguagem, shell, *framework*) utilizada para desenvolver o sistema;
- *Entrada em produção do sistema* - indica se o sistema chegou a entrar em produção, comercial ou não. Alguns sistemas foram desenhados apenas como ferramentas pedagógicas, outros nunca

passaram de protótipos e outros chegaram a ser utilizados em obras. É importante referir que não foi possível obter esta informação para alguns dos SP, nesses casos esta rubrica toma o valor *desconhecido*;

- *Adições / Adjuvantes* - assinala se contempla adições ou adjuvantes no betão, um importante indicador pois hoje em dia os betões quase sempre possuem pelo menos um tipo destes constituintes na sua composição;
- *Normas* - quais as normas legais tidas em conta na produção do betão; cada país tem a sua, que pode ser comum a outros (existem normas europeias, por exemplo). Uma característica importante pois se a norma não for a adequada para um determinado país, o betão não é conforme;
- *Betão suportado* - existem vários tipos diferentes de betão, dependendo das características requeridas; nesta coluna são indicados os suportados.

É de referir também que foi acrescentada uma linha para o sistema pericial proposto, de modo a auxiliar na comparação.

**Tabela 4.1:** Comparação entre os sistemas para estudos de composição de betão analisados

Sistema Pericial	Ferramenta utilizada	Entrou em produção?	Adições / Adjuvantes	Normas	Betão suportado
BETVAL	Insight 2+	apenas para aprendizagem	não	Finlândia?	normal, pronto
COMIX	desconhecido	desconhecido	não	Nova Zelândia	normal
CONCEX	Rulemaster, C, Fortran	desconhecido	não	EUA	normal
Concrete Mix Designer	Personal Consultant Plus, Basic	apenas para aprendizagem	não	EUA	normal
EXMIX	Level 5	desconhecido	não	Canadá	normal
KBES for Concrete Mix Design	Exsys, Quattro Pro	desconhecido	sim	EUA	leve, normal, pesado
HPCMIX	Kappa-PC	protótipo	sim	Malásia	alta densidade
Application of Aut. Acq.	Visual C++	desconhecido	sim	China	normal
ESBAC	Kappa-PC	protótipo	não	Bangladesh	agreg. tijolo reciclado
<b>SISTEMA PROPOSTO</b>	<b>Drools, C#, VB.net</b>	<b>protótipo</b>	<b>sim</b>	<b>Europeias, PT</b>	<b>normal, auto-compactável,</b>
<b>(Betacomp)</b>					<b>elev. e elevadíssimo desempenho</b>
<i>Outros sistemas</i>					
Folhas de cálculo	Excel ou equivalente	sim	não aplicável	não aplicável	não aplicável
MATLAB program for Diag.	MATLAB	sim	sim	EUA	agregados leves
MATWIN	Access	sim	desconhecido	PT	normal
SIKACOMP	VB.NET	sim	sim	PT	normal

### 4.3 Necessidade de soluções

Para se desenvolver uma nova aplicação, deve haver uma necessidade a satisfazer. Neste ponto são apreciados vários aspectos das soluções apresentadas no capítulo anterior, no sentido de esclarecer o porquê de desenvolver um novo sistema pericial, os quais são expostos abaixo.

- **Exequibilidade das soluções desenvolvidas**

Muitas das soluções encontradas nunca chegaram a passar de protótipos ou são soluções académicas, que não obstante serem perfeitamente válidas, nunca tiveram uma utilização num ambiente de trabalho real, pelo menos em números significativos.

- **Conformidade das normas do betão**

Existem outras limitações em vários dos sistemas periciais desenvolvidos, como é referido em [Abdullahi et al., 2008]. Alguns deles usam por base normas antigas para a especificação do betão, estando já desactualizados. Recorrendo a um exemplo concreto, em Portugal até há alguns anos atrás eram utilizadas séries de peneiros baseadas em unidades de medida americanas para a análise granulométrica, neste momento essas mesmas medidas estão de acordo com as normas europeias e o seu cumprimento é obrigatório. Isto significa que um software, por mais eficaz que seja, se não for actualizado tem toda a sua funcionalidade comprometida.

Além disso, as normas nacionais de muitos países não são contempladas, como por exemplo as aplicadas em Portugal. Este facto por si só já justificaria a criação de um novo sistema, pois nenhum dos SP analisados atrás contempla explicitamente as normas portuguesas actuais (as quais por sua vez são baseadas nas europeias). Se considerarmos o restante software, temos que o *Sikacomp* cumpre na íntegra estas normas, pelo que é utilizado ainda hoje em dia; O *Matwin* está pensado para o mercado nacional, mas sendo já datado de 1999 é altamente improvável que cumpra na totalidade as normas actuais (não foi possível ter acesso a uma cópia do programa) - pois foram realizadas alterações às normas utilizadas em Portugal na década seguinte.

- **Suporte aos constituintes do betão**

Após observação da tabela 4.1, é possível verificar que apenas três dos nove sistemas periciais analisados suporta adições e/ou adjuvantes na composição do betão. Ora, hoje em dia é muito difícil um betão não possuir pelo menos uma adição ou um adjuvante para melhorar as suas características (muitas vezes ambos e não é raro um betão ter dois adjuvantes e uma adição, por exemplo). Destes três, apenas dois são pensados para betão normal e nenhum deles suporta as normas europeias ou portuguesas. Este facto abre espaço para novas soluções.

- **Domínio de conhecimento da aplicação**

Em outros SP verifica-se que têm um âmbito de aplicação muito restrito, limitando a sua utilização a alguns nichos. Por exemplo, alguns dos sistemas desenvolvidos apenas são aplicáveis a determinados tipos de betão, como os betões de agregados leves, betões com agregados reciclados ou betões de elevado desempenho.

- **Suporte a sistemas operativos**

Inevitavelmente, a aplicação desenvolvida, seja ela qual for, terá de ser instalada num computador pessoal - dado que nenhum dos sistemas analisados foi desenvolvido para a *web*. Se não tiver suporte para um sistema operativo (SO) moderno, como o *Microsoft Windows*, *MacOS X* ou uma distribuição *Linux*, dificilmente será utilizada. Se bem que uma parte destes sistemas tenha chegado a ter aplicação prática em locais de obra, trata-se na sua esmagadora maioria de software bastante antigo e que dificilmente poderá ser executado em SO actuais.

- **Outros aspectos a considerar**

Numa outra categoria de soluções existentes, alternativas aos sistemas periciais, como por exemplo a utilização das folhas de cálculo, existem outros tipos de questões. Como são normalmente realizadas pelo próprio utilizador final, por regra não estão pensadas para serem usadas por outras pessoas. Por outro lado, são bastante estáticas e limitam-se a realizar cálculos, não possuindo a flexibilidade e a lógica inerentes a um sistema pericial bem concebido. O software desenvolvido em *Matlab*, apesar de ter características interessantes, apenas suporta as normas americanas. Conforme foi referido, o *Matwin* do LNEC é um software já desactualizado, consideravelmente oneroso (custa 2500€+ IVA), e foi desenvolvido para Windows 9x [Laboratório Nacional de Engenharia Civil, 2012].

Em suma, podemos arriscar dizer que para a realidade nacional - à excepção do *Sikacomp 2.5* - não existem sistemas disponíveis, periciais ou não, que contemplem a realidade nacional e as normas legais em vigor, portuguesas ou europeias, e que possam ser utilizadas num sistema operativo comum, numa versão actual. Por outro lado, o suporte a adições e adjuvantes é realizado apenas por uma minoria, e este é um aspecto fulcral na composição de betões actuais.

Acima de tudo, e salvo a excepção referida, nenhum dos sistemas atrás expostos cumpre todos estes requisitos na totalidade. Daí surge o propósito deste sistema pericial agora desenvolvido, no âmbito desta dissertação.

## 4.4 Objectivos e relevância da aplicação

A composição de um betão é um estudo complexo, onde é necessário satisfazer diferentes necessidades, por vezes antagónicas, como obter boa trabalhabilidade e simultaneamente uma resistência adequada do produto final, um custo admissível e suportável pelo produtor do betão, e ainda fazer cumprir as limitações impostas pelas normas nacionais e europeias em vigor. As variáveis são diversas e pressupõem várias precedências entre si, criando relações intrincadas entre os diferentes aspectos a considerar.

Por outro lado, o conhecimento dos peritos sobre a criação de um betão, embora tenha uma sólida base teórica, é em grande parte empírico, o que torna mais difícil a transmissão integral desse saber para outra pessoa ou a sua integração em software.

A aplicação desenvolvida propõe-se utilizar uma nova aproximação, através da implementação de um sistema pericial com características um pouco diferentes dos anteriores. Através da utilização de ferramentas de desenvolvimento mais actuais e com suporte a regras de *Business Intelligence*, almeja solucionar alguns dos problemas actuais dos profissionais e peritos do betão na composição deste material, através de uma abordagem mais generalista no que toca ao âmbito da sua aplicação.

Tem por finalidade principal realizar a definição da composição do betão, ou seja, especificar quais os seus constituintes e quais as suas proporções e quantidades, do modo mais eficiente possível, de acordo com o conhecimento pericial recolhido. Em termos gerais, pretende alcançar os seguintes objectivos:

- com a assistência do programa, substituir parcialmente o perito humano, ao permitir que alguém atinja níveis de desempenho próximos aos de um especialista. O programa acompanha o utilizador, perguntando quais as especificações do betão e providenciando um solução de composição do betão;
- no caso de o utilizador ser já um perito, melhorar a sua produtividade, sendo possível produzir resultados mais rapidamente. O processo é mais célere que o habitual pois as decisões são tomadas pelo sistema em fracções de segundo, evitando alguns passos iniciais de ponderação por parte do perito;
- dotar a ferramenta de capacidade pedagógica, com a facilidade de criação de diferentes cenários alternativos para análise e experimentação, assim como explicações sobre as opções tomadas pelo sistema.

A aplicação suportará os seguintes tipos de betão:

- betão normal;
- betão de elevado desempenho;
- betão de elevadíssimo desempenho;

- betão auto-compactável.

Ou seja, não estará focada apenas em nichos, mas antes em alguns dos tipos de betão mais correntes. É no entanto importante referir que existem outros tipos, mais específicos, que não são contemplados. Como é possível ver na tabela 4.1, nenhum dos sistemas apresentados cobre todos os tipos de betão, pois isso obrigaria a utilizar metodologias muito distintas para a composição e essa abrangência seria difícil de atingir.

É também adaptada à realidade portuguesa, pelo que foram ouvidos peritos portugueses antes e durante o seu desenvolvimento. As normas legais em vigor, nacionais e europeias, são também consideradas.



## Capítulo 5

# Sistema Pericial Desenvolvido

*“Le matériel se rit des équations où nous pensons pouvoir le renfermer quand éventuellement nous oublions de l’essayer tant de fois qu’il faut.”*

— M. Duriez, *Cours de Matériaux de Construction, Paris, École Nationale des Ponts et Chaussées (1944).*

A solução proposta para a criação de estudos de composição de betão é na sua essência um sistema pericial baseado em regras, baptizado de *Betacomp* e composto por três componentes principais: a *Interface do sistema pericial*, o *Módulo de cálculo de composição de betão* e a *Base de conhecimento com as regras de produção*.

Neste capítulo será explicada a solução encontrada e respectiva fundamentação, assim como os seus diferentes componentes e o seu modo de utilização.

### 5.1 Tecnologias relevantes

Um dos primeiros passos no desenvolvimento do sistema pericial foi a escolha da ferramenta a utilizar na sua programação. Esta foi uma fase que demorou algum tempo pois a escolha não foi óbvia, tendo implicado a pesquisa de diferentes soluções até à redução a um pequeno grupo de hipóteses mais fortes.

#### 5.1.1 A escolha de um paradigma

Existem vários paradigmas de programação para a criação de sistemas periciais. Os sistemas mais modernos, no entanto, usam na sua grande maioria regras de produção ou *business rules*.

Um sistema de regras permite separar a lógica inerente a um problema, da componente funcional da aplicação. Este código fica separado das regras, a lógica do sistema. Podemos deste modo focar-nos na

resolução de uma questão, considerando apenas a sua lógica, sem a necessidade de se investir demasiado tempo com constrangimentos técnicos - focando-nos essencialmente no conhecimento.

A separação da lógica de uma aplicação do seu código pode permitir aumentar a eficiência do software. Mas convém ter presente que no caso de uma aplicação de software de pequena dimensão, a utilização de um sistema de regras pode porventura até torná-la mais complexa. Já se considerarmos uma aplicação com uma certa dimensão, e com uma lógica de negócio volátil, a sua manutenção torna-se muito mais complicada e onerosa em termos de tempo e da ocupação de recursos humanos necessários ao seu bom funcionamento - neste tipo de situações temos o cenário ideal para a implementação de um sistema baseado em regras.

O estudo prévio realizado para a escolha da plataforma a utilizar para o desenvolvimento do SP, teve como consequência prática a exclusão dos sistemas que não seguissem o paradigma das regras de produção e, por conseguinte, a separação entre conhecimento e mecanismos de processamento desse conhecimento.

### 5.1.2 Ferramentas de desenvolvimento consideradas

Foram portanto considerados alguns dos principais sistemas para desenvolver sistemas periciais modernos, mais concretamente o Jess, CLIPS, Prolog e Drools.

Existem muitas variantes destes, dentro do CLIPS temos, para além do *framework* original, outras variantes como o *WebCLIPS* (uma implementação de CLIPS como uma aplicação CGI, orientada para a *web* [Giordano, 2003]), o *FuzzyCLIPS* (com suporte adicional a regras e factos utilizando Fuzzy Logic [Orchard, 1998]), o *KAPICLIPS* (o qual implementa uma arquitectura com agentes através da interface *Knowledge Query and Manipulation Language API* (KAPI) [Friedman-Hill, 1995]), entre outros.

O próprio Jess é uma implementação de CLIPS em Java, mas dado que já possui características distintas na sua implementação foi considerado numa categoria à parte.

Relativamente ao Prolog, o cenário é semelhante. Temos o *YProlog*, uma implementação em Java desta linguagem que permite que a base de conhecimento Prolog seja acedida como uma base de dados SQL [Schooten, 2006], o *tuProlog*, também uma implementação em Java realizada pela Universidade de Bolonha para aplicações distribuídas e que suporta nativamente vários paradigmas de programação, sendo possível integrar com plugin para *Eclipse IDE*, com a plataforma *.NET* da *Microsoft* e até mesmo *Android* [APICe, 2011]. Uma distribuição, a *Prolog.NET*, como o próprio nome indica, foi desenvolvida para integração com plataforma *.NET* da *Microsoft* [Hodroj, 2011]. Existe inclusivamente um implementação escrita em linguagem *C* da Universidade do Porto, o *YAProlog*, optimizado para alto desempenho [Advanced Computing Systems, 2010].

Na realidade existe uma enorme proliferação de variantes destas linguagens, provavelmente dezenas, pelo que não fará sentido expor individualmente cada uma.

O Drools é um *rule engine* orientado a objectos, também conhecido como um *production rule system* ou sistema de regras de produção. Na última versão é já designado de *Business Logic integration Platform*. Actualmente existe uma distribuição suportada e oficial da *Red Hat*, onde toma a designação *JBoss Rules*, mas existe como solução *open source*. Fazendo uma comparação um pouco livre com os sistemas operativos, é semelhante à versão *CentOS* do *Red Hat Linux*. Existe ainda uma versão, igualmente *open source*, para a plataforma *.NET*: o *Drools.NET*.

O ponto que se quer defender com esta argumentação, é que embora estejam a ser comparadas quatro plataformas, na realidade podemos considerar que são bastante mais dado que estes são os sistemas “pai” de uma família de produtos.

Outras ferramentas de desenvolvimento para SP foram tidas em conta durante a fase de pesquisa, mas devido à sua menor utilização e ao facto de não trazerem vantagens significativas sobre as acima mencionadas, não foram consideradas. Entre estas estão por exemplo a *d3web* [d3web, 2011], *G2 2011* [Gensym, 2012] e o *OpenKBM* da *Rocket Software*.

Foi construída a tabela 5.1 que resume a informação recolhida e tida em conta para suportar a escolha da ferramenta:

**Tabela 5.1:** Comparação entre algumas das principais ferramentas de desenvolvimento de sistemas periciais

	<b>Jess</b>	<b>CLIPS</b>	<b>PROLOG</b>	<b>Drools</b>
<b>Inferência</b>	forward e backward chaining	forward chaining	backward chaining	forward chaining
<b>Interface com utilizador</b>	command-line, GUI apps, servlets, applets	command-line, GUI	command-line, GUI	command-line, GUI apps, servlets, applets
<b>Pattern-Matching</b>	Rete	Rete	Unification	ReteOO
<b>Optimização</b>	para desempenho, maior ocupação da memória RAM	para desempenho, maior ocupação da memória RAM	para o espaço em memória, em detrimento do desempenho	para desempenho, maior ocupação da memória RAM
<b>Plataforma</b>	Java	shell CLIPS	shell Prolog	Jboss - Java
<b>Integração com Visual Studio</b>	não	não	sim ( <i>Prolog.NET</i> , <i>tuProlog</i> )	sim ( <i>Drools.NET</i> )
<b>Tipo Licença</b>	comercial	<i>open source</i>	<i>open source</i>	<i>open source</i>

A escolha de uma plataforma para desenvolvimento do *Betacomp* não foi uma tarefa fácil. Todos os sistemas analisados tinham pontos fortes a seu favor, pelo que foi adoptado um método de exclusão dos candidatos, baseado essencialmente nos seguintes critérios:

- **Software gratuito** - determinou-se que seria dada primazia a uma solução *open source* para o desenvolvimento do SP. Este parâmetro excluiu o Jess à partida, pois é um software comercial;
- **Algoritmo *pattern-matching*** - a eficiência do algoritmo foi considerado um ponto fundamental para a escolha;
- **Tipo de inferência** - para o tipo de SP a desenvolver, foi dada preferência à lógica de *forward chaining*, é a que mais se assemelha ao raciocínio efectuado pelos peritos da área (embora não fosse obrigatório);
- **Integração com a plataforma .NET** - seria interessante ter integração com esta plataforma, dado que o módulo de cálculo de composição do betão foi baseado num já existente (*Sikacomp 2.5*), desenvolvido em *VB.NET*;
- **Interface gráfica apelativa** - uma interface com boa qualidade gráfica, quer para o desenvolvimento, quer para o utilizador final, é um aspecto importante a considerar.

O algoritmo *Rete* é considerado sem grande controvérsia como sendo bastante eficiente para a maioria das situações, e o Drools utiliza uma versão melhorada do mesmo - o *ReteOO* [Bali, 2009]. Tendo em consideração todos os critérios apresentados, a escolha que pareceu mais lógica recaiu no Drools, que os satisfazia em pleno - mais especificamente, no *Drools.NET*, a versão .NET desta plataforma, desenvolvida pela Sourceforge [Sourceforge, 2006].

O *Drools.NET* possibilita implementar um sistema de regras juntamente com um motor de inferência e, em adição, disponibiliza uma interface com o Microsoft Visual Studio .NET, tendo a programação de ser realizada na linguagem *C#* (C Sharp). Por este motivo temos que o *Betacomp* foi programado em *VB.NET*, *C#* e a sua base de conhecimento em regras de produção Drools.

### 5.1.3 Outras ferramentas utilizadas

Para além das ferramentas já referidas, a um outro nível outras foram de grande importância no desenvolvimento desta dissertação e do próprio sistema pericial. Uma breve lista é enunciada:

- Microsoft Visual Studio 2008 - desenvolvimento em *VB.NET* e *C#*;
- Eclipse IDE Europa - desenvolvimento de regras Drools;
- TeXnicCenter 1.0 RC1 - IDE para desenvolvimento em LaTeX;
- MiKTeX 2.9 - suporte a LaTeX;
- JabRef 2.7.2 - suporte bibliográfico.

O *Drools.NET*, conforme foi dito, permite a utilização do Drools na plataforma .NET. Tem por sua vez, os seguintes requisitos:

- *Microsoft .NET Framework 2.0* ou superior;
- *IKVM 0.28.0.0* - uma implementação de Java para o *Microsoft .NET Framework*, o qual é incluído na distribuição *Drools.NET* [Frijters, 2011];
- *NUnit-2.2.6* ou superior, um *unit-testing framework* semelhante ao *JUnit* para as linguagens .NET, também incluído na distribuição [Poole et al., 2007].

## 5.2 Arquitectura do sistema

Nesta secção é oferecida uma visão geral da arquitectura do *Betacomp* para uma melhor compreensão do sistema. A sua concepção, constituintes e o fluxo de invocação dos seus módulos, assim como da informação, são aqui descritos.

### 5.2.1 Módulos do *Betacomp*

Existem três módulos principais, todos interligados, cada qual por sua vez constituído por diferentes componentes. Foi criada em *Visual Studio 2008* uma *Solution* intitulada *Betacomp-with-Drools.sln*, a qual contém dois projectos - um em *C#* e outro em *VB.NET*.

Esta estrutura foi necessária pois não é possível juntar código de duas linguagens diferentes no mesmo projecto do *Visual Studio*. Paralelamente, existe um ficheiro de regras do tipo *drl* com as regras de produção.

Temos portanto os seguintes módulos:

1. **Interface do sistema pericial** - a interface onde se inicia o estudo de composição de betão e onde o utilizador recebe as sugestões realizadas pelo SP. É implementada através do projecto *Betacomp\_SP*, o qual foi desenvolvido em *C#* (ver figura 5.1). Os seus principais componentes são:
  - *Classe Estudo* - classe com a informação necessária ao estudo de composição de betão, uma instância é criada no início da aplicação e em seguida é passada ao sistema pericial; quando as decisões do SP são apresentadas, este objecto é retornado para que a informação seja utilizada pelo **Módulo de cálculo de composição de betão**.
  - *Classes de Especificação do Betão* - classes com as diversas especificações do betão, que agem como uma base de dados estática que representa as diversas normas e especificações do betão; estas irão alimentar a base de conhecimento do SP durante o seu funcionamento.

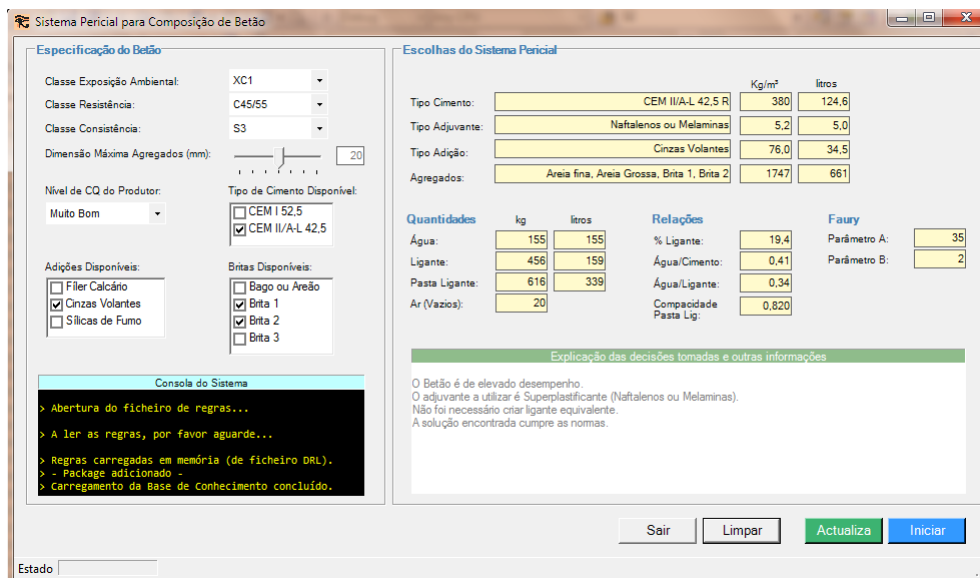


Figura 5.1: A interface do sistema pericial - projecto *Betacomp\_SP* em C#

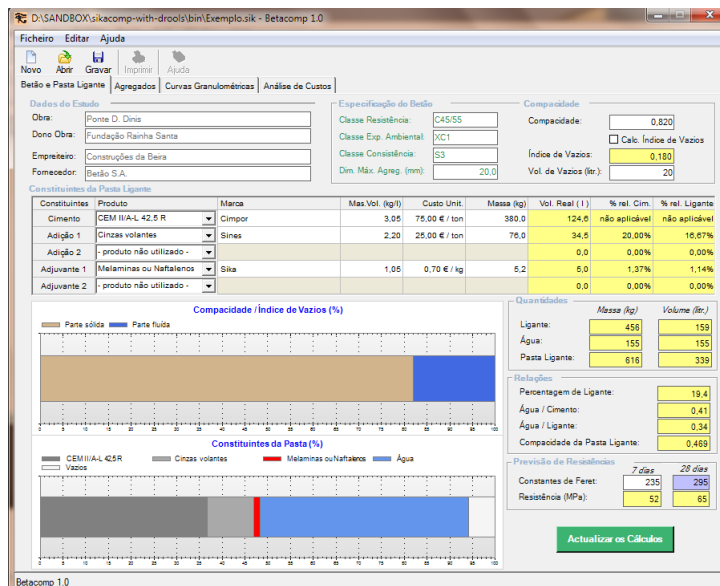


Figura 5.2: O ecrã principal do módulo de cálculo de composição - projecto *BetacompCalc* em VB.NET

2. **Módulo de cálculo de composição de betão** - após estarem definidos os constituintes do betão, este módulo trata dos cálculos que devem ser efectuados e fornece mais ferramentas de apoio à decisão.

Estas são submódulos da aplicação, como a análise granulométrica, a análise gráfica dos agregados e respectiva curva das granulometrias e ainda uma análise de custos. É implementada através do projecto *BetacompCalc*, escrito em *VB.NET* (ver figura 5.2). A sua principal funcionalidade está representada pelas classes:

- *Classe EstudoMain* - por sua vez agrega as classes *EstudoInputs* e *EstudoOutputs*, e representa o conhecimento relativo ao estudo. Engloba o conhecimento retornado pela *Classe Estudo* atrás referida na **Interface do sistema pericial**, mais os valores adicionais inseridos pelo utilizador (*EstudoInputs*) e o resultado dos cálculos realizados (*EstudoOutputs*).
- *Classes GridsAgregados, GridsConstituintes, SeriesPeneiros, CalculosFaury* - classes para a implementação das grelhas de introdução de dados presentes neste módulo, assim como para possibilitar a representação gráfica dos constituintes e realizar os cálculos necessários.

3. **Base de conhecimento com as regras de produção** - está representada no ficheiro de texto *DroolsRules.drl*, no formato *drl* do Drools (*Drools Rule Language*) e contém as regras de produção do SP. As regras e os seus formatos são desenvolvidos em mais pormenor no ponto 5.4.

## 5.2.2 Interligação dos módulos

O projecto *Betacomp\_SP* é o primeiro a ser compilado e gera uma *dll*<sup>1</sup> (*SistemaPericial.dll*), a qual é utilizada como um recurso pelo projecto *BetacompCalc* (escrito em *VB.NET*) - de modo a implementar a Interface do SP (escrita em *C#*).

Por sua vez, o projecto *Betacomp\_SP* tem uma referência ao ficheiro de regras *DroolsRules.drl*, de modo a poder utilizá-lo para alimentar a base de conhecimento do sistema pericial com as regras aí presentes. O projecto *Betacomp\_SP* (escrito em *C#*) está preparado para interagir com as regras Drools, através do suporte providenciado pelo *Drools.NET*.

Estas relações estão representadas graficamente na figura 5.3.

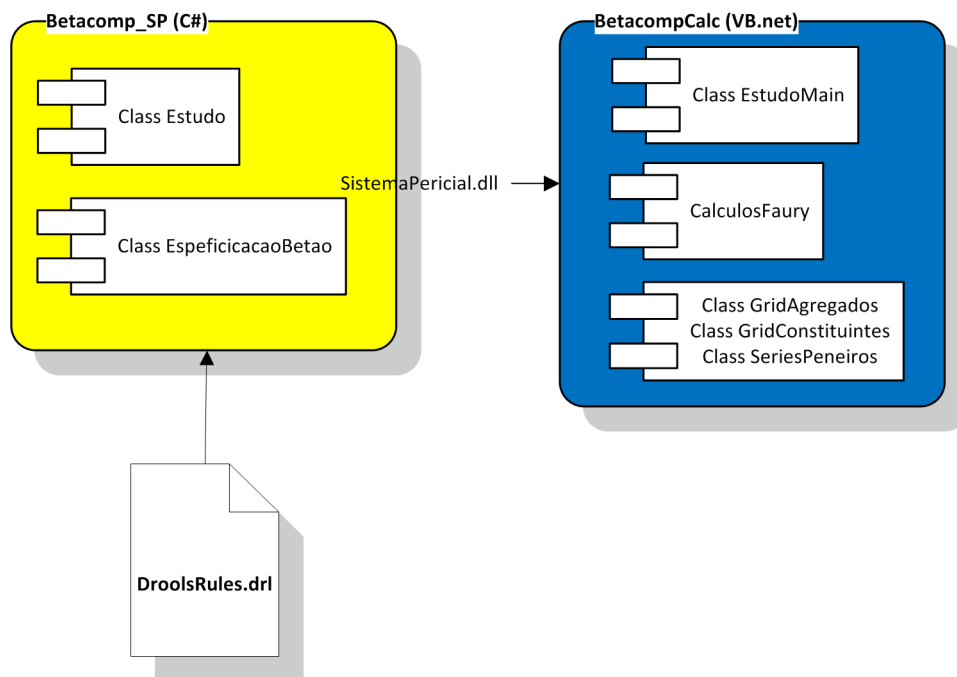
## 5.2.3 Fluxo de invocação

Recapitulando, o módulo inicial é portanto o *BetacompCalc*, o qual invoca a interface do SP (*Betacomp\_SP*), que por sua vez utiliza a base de conhecimento do ficheiro *DroolsRules.drl*.

O modo como estas invocações são realizadas segue a seguinte sequência:

---

<sup>1</sup>*Dynamic-link library*



**Figura 5.3:** Integração dos componentes e principais classes do *Betacomp*

1. O *BetacompCalc* é o módulo de arranque do *Betacomp* e inicia a sua execução.
2. O ficheiro *SistemaPericial.dll* com a implementação da interface do sistema pericial (*Betacomp\_SP*) é utilizado como recurso do módulo *BetacompCalc*, de modo a permitir a utilização do código C# do SP.
3. O sistema Pericial inicializa as suas classes e o módulo *Betacomp\_SP* carrega as regras Drools a partir do ficheiro *DroolsRules.drl*, que por sua vez cria uma nova *dll* para utilização do SP - nomeada *CompiledRules0.dll* - com as regras que constituem a base de conhecimento. A partir deste momento o sistema está pronto a ser utilizado.
4. A qualquer momento o utilizador pode sair da **Interface do sistema pericial** e voltar ao **Módulo de cálculo de composição de betão** e vice-versa.
5. No **Módulo de cálculo de composição de betão**, quando for necessário, o utilizador pode gravar o estudo que realizou em ficheiro ou carregar um estudo previamente realizado. Estes são gravados com a extensão *.sik*.

## 5.3 Aquisição do conhecimento

O conhecimento adquirido para o sistema pericial desenvolvido é essencialmente proveniente de um perito que não só detém o domínio teórico e académico de longa data, como possui vasta experiência na composição de betões, um reconhecido profissional e consultor de algumas das mais importantes empresas da área da construção há várias décadas. Este precioso contributo ficou a cargo de Jorge Lourenço, Professor adjunto do ISEC no Departamento de Engenharia Civil. Foi igualmente muito importante a co-operação para este trabalho com outras pessoas que trabalhavam directa ou indirectamente com a feitura do betão, a diferentes níveis.

### 5.3.1 O conhecimento pericial

Para a aquisição do conhecimento do perito, foram realizadas várias e longas sessões de trabalho, habitualmente no Laboratório de Engenharia Civil do Instituto Superior de Engenharia de Coimbra (ver figura 5.4). Distribuídas ao longo de meses, estas sessões demonstraram ser tão extensas como produtivas, e lentamente foi-se construindo uma base de conhecimento - inicialmente informal e que com o passar do tempo foi sendo refinada e elaborada, graças à compreensão do perito. Para além do seu largo conhecimento teórico, o verdadeiro desafio terá sido possivelmente a tradução do seu saber empírico em regras lógicas e estruturadas, que pudessem ser utilizadas no sistema pericial.

Este foi provavelmente o trabalho mais exaustivo mas também o mais enriquecedor deste estudo.



**Figura 5.4:** O Laboratório de Eng. Civil do ISEC, local das sessões de aquisição de conhecimento

### **5.3.2 As normas legais a respeitar**

Para além da aquisição do saber simultaneamente teórico e empírico do perito, foi igualmente tido em consideração um outro tipo de conhecimento, este mais formal, que consiste das normas portuguesas em vigor: as norma NP EN 206-1 e E464, às quais deve obedecer qualquer betão produzido em Portugal.

A norma NP EN 206-1 estabelece a especificação, desempenho, produção e conformidade do betão [APEB - Associação Portuguesa das Empresas de Betão Pronto, 2008] no nosso país, sendo por sua vez baseado em normas europeias.

A especificação E464 por seu lado define os requisitos necessários para o betão resistir às acções ambientais, ou seja, os valores limite para a composição e propriedades estabelecidas para o betão [LNEC, 2007]. Aqui são definidas diversas classes de exposição ambiental, como a exposição a elementos químicos, corrosivos ou ao gelo/degelo, entre outros.

As regras do SP expressam os requisitos e valores limites destas normas.

### **5.3.3 Método de aquisição do conhecimento**

Para adquirir e conseguir traduzir no formato de regras o conhecimento do perito foi necessário seguir uma metodologia, que embora não fosse rígida, se baseou nas seguintes etapas:

1. *Identificação da questão* - quais os dados com que seria necessário lidar, as restrições obrigatórias a cumprir e que critérios utilizar;
2. *Conceptualização* - tentar traduzir as relações entre os diferentes componentes do problema, de modo a perceber o fluxo e a ordem da informação e o modo como interagem, para em seguida criar as estruturas necessárias à futura implementação;
3. *Implementação* - Expressão em regras da conceptualização previamente realizada;
4. *Teste* - verificação dos resultados, o mais exaustiva possível, testando a sequência e os valores apresentados pelo SP.

## **5.4 Representação do conhecimento**

### **5.4.1 Anatomia de uma regra**

Nesta secção examinamos as regras Drools, de modo a compreender os seus componentes. Estas seguem o formato apresentado abaixo:

```

rule ‘‘nome da regra’’
<atributos opcionais>
  when
    <LHS>
  then
    <RHS>
end

```

onde:

- *nome da regra* - identificação da regra;
- *<LHS>* (*Left-Handed Side*) - o lado esquerdo da regra, onde se encontram as condições que terão de ser satisfeitas para que a regra seja aplicada;
- *<RHS>* (*Right-Handed Side*) - o lado direito da regra, as acções a realizar caso as condições no *<LHS>* se verifiquem;
- *<atributos opcionais>* - atributos que pode ser utilizados para controlar a execução das regras, os quais são descritos em mais pormenor no ponto 5.5.

Um exemplo prático da utilização de uma regra na base de conhecimento do *Betacomp* é mostrado abaixo, a título demonstrativo:

```

package SistemaPericial
import SistemaPericial.Classes

rule "Teor Minimo Ar Obrigatorio"
salience -10
  when
    estudo: Estudo( ExpAmbiental >= 13, ExpAmbiental <= 15) #XF2 a XF4
  then
    SistemaPericial.frmSP.outputExplicacao("Teor minimo de ar obrigatório pois é um betão XF."
    estudo.TeorMinAr = 4;
end

```

As duas primeiras linhas são um prelúdio às regras, mas importantes para percebermos como a base de conhecimento está estruturada. Na primeira linha de código temos a indicação de um *package*, que realiza a função de um *namespace*<sup>2</sup> para as regras do ficheiro [Bali, 2009]. A linha imediatamente a

<sup>2</sup>um espaço para evitar a ambiguidade de nomes na programação

seguir contém o comando *import*, equivalente a *using* ou *include* em outras linguagens, permite utilizar classes definidas na interface do sistema pericial.

A regra em si traduz-se em que se um betão pretendido está entre as classes de exposição ambiental “XF2” a “XF4”, terá de ter um teor mínimo de ar de 4% e alerta o utilizador para o facto com uma mensagem.

Como é possível verificar, as regras podem ser compreendidas com uma certa facilidade, mesmo para quem não tenha muita afinidade na área, permitindo que o próprio perito possa validar o conhecimento descrito pelas regras.

## 5.4.2 Base de conhecimento

O conhecimento fundamental do sistema pericial desenvolvido está na forma de regras, armazenadas num ficheiro com a extensão *.drl*, as quais seguem a configuração descrita no ponto anterior. Este é o pilar de todo o sistema, a base para a tomada de decisões do motor de inferência e o conhecimento necessário para definir a composição de um betão.

Dado que o número de regras não era demasiado extenso, foi criado um único ficheiro de regras - o *DroolsRules.drl*. A utilização deste ficheiro de regras pode ser muito facilmente alterada para um outro, ou incluir novos ficheiros. Esta é uma das grandes vantagens do Drools, a opção de poder alterar de um modo modular conjuntos de regras relacionadas, de modo a facilitar a sua manutenção e compreensão. Para esse efeito, bastaria alterar duas linhas de código, no projecto *Betacomp\_SP*:

```
stream = new FileStream(@".\DroolsRules.drl", FileMode.Open, FileAccess.Read);
PkgBuilder.AddPackageFromDrl("SistemaPericial.DroolsRules.drl", stream);
```

Para acrescentar mais ficheiros de regras, o que pode ser útil por uma questão de organização (um ficheiro para cada tipo de regras por exemplo), bastaria acrescentar mais tantas linhas quantos os ficheiros necessários. No final, o Drools irá considerar as regras de todos os ficheiros como uma única base de conhecimento:

```
stream = new FileStream(@".\DroolsRules.drl", FileMode.Open, FileAccess.Read);
stream2 = new FileStream(@".\MoreRules.drl", FileMode.Open, FileAccess.Read);
...
PkgBuilder.AddPackageFromDrl("SistemaPericial.DroolsRules.drl", stream);
PkgBuilder.AddPackageFromDrl("SistemaPericial.MoreRules.drl", stream2);
...
```

É inclusivamente possível aceder a ficheiros *drl* a partir de um URL, permitindo o acesso a regras de um modo muito dinâmico de qualquer localização na *web*. No entanto, no presente projecto tal funcionalidade não foi necessária.

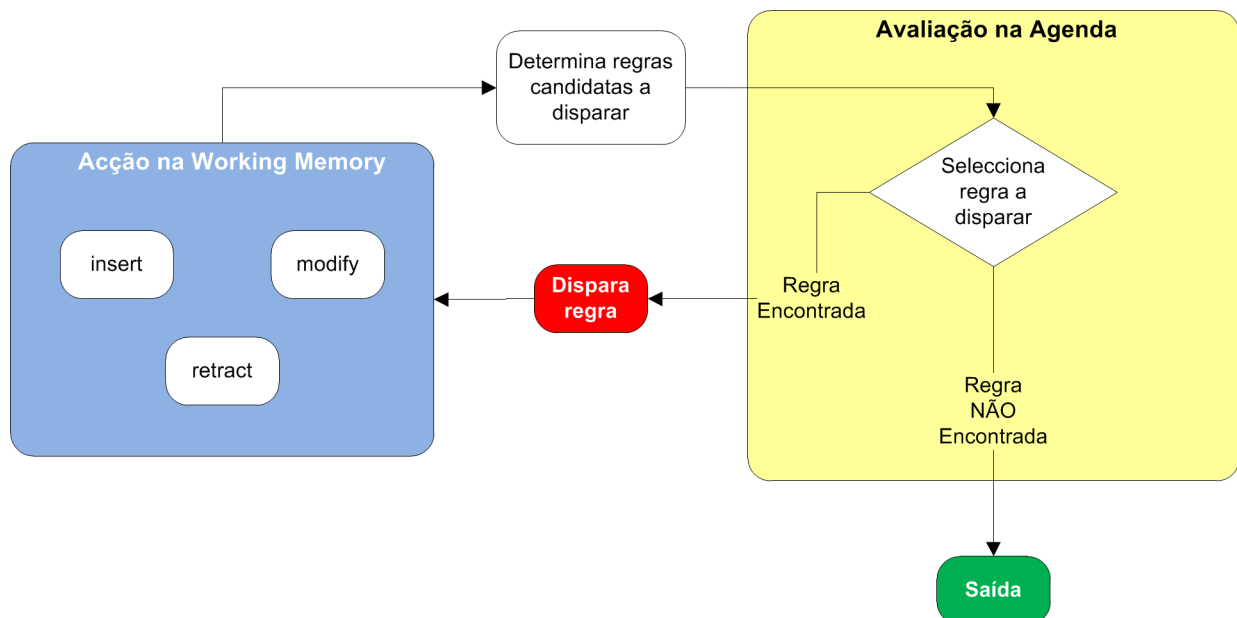
A base de conhecimento foi construída dando primazia a duas características essenciais:

- **simplicidade das regras** - dando preferência à criação de duas regras mais simples em vez de uma só com mais condições ou acções; para além da facilidade de compreensão, deste modo pretende-se atingir uma versatilidade maior das regras, pensando na sua expansão futura.
- **controlo da ordem de execução** - a escrita das regras teve especial atenção ao modo como podem afectar os resultados, conforme a sua ordem de disparo. As opções tomadas são desenvolvidas no ponto 5.5.

## 5.5 Raciocínio

Na criação das regras, foi claro ao final de algum tempo de desenvolvimento, e também de um processo de tentativa-erro, que seria necessário modificar a ordem de execução das mesmas em alguns casos. Este método de alteração da ordem de disparo das regras activadas é habitualmente designado por *Conflict Resolution*, o qual pode aplicar-se quando há mais de uma regra activada num dado momento.

Por defeito, o Drools utiliza uma lógica LIFO para o disparo das regras, entre outras estratégias - a figura 5.5 ilustra uma visão de alto-nível do processo. No entanto, em certas situações foi necessário recorrer explicitamente a diferentes mecanismos de modo a conseguir a correcta execução das regras, os quais são explicitados de seguida.



**Figura 5.5:** A execução de regras no Drools [Proctor, 2010]

### 5.5.1 Resolução de conflitos e proeminência

Em regras Drools, o atributo *saliency* (ou proeminência) é representado por um número inteiro que tem por defeito o valor zero (caso não seja definido) e em que quanto maior for o seu valor, maior a prioridade da regra em causa na fila de disparo [Proctor et al., 2005].

No presente caso, foi atribuída uma prioridade inferior às regras que realizam validações finais (*saliency -10*) e um valor mais alto às regras que realizam verificações que têm forçosamente que ser realizadas logo no início da execução do SP, pois outras regras e resultados dependem directamente do resultado destas (*saliency 10*).

Deste modo foi dada a primazia de execução às regras mais críticas e uma prioridade inferior às restantes. Foi ainda criado um outro nível - *saliency -5* - para regras relacionadas com a apresentação de mensagens de explicação ou descrição de opções do utilizador. Resumidamente, existem quatro níveis de prioridade nas regras, em ordem crescente de relevância:

1. *saliency -10*, para as regras menos críticas e que devem ser executadas mais tarde, como validações finais;
2. *saliency -5*, regras relacionadas com mensagens de texto destinadas aos utilizadores, como as explicações;
3. *saliency 0*, ou seja, o atributo não é definido (prioridade normal);
4. *saliency 10*, regras mais relevantes, cuja execução é prioritária.

### 5.5.2 Prevenção de ciclos recursivos

Outro atributo importante para a correcta execução das regras é o *no-loop*. Quando uma regra é executada, por vezes um facto é modificado - alterando o estado do sistema - e este pode despoletar a sua execução novamente, dando origem a recursividade e potencialmente um ciclo infinito. Este atributo é do tipo *boolean* e tem por defeito o valor *false*, mas se colocado no início de uma regra (logo passando a *true*), previne este tipo de situação ao ignorar uma segunda chamada à regra.

A sua utilização é demonstrada no código seguinte, onde no *<RHS>* (ou seja, na consequência da regra) há uma alteração do objecto *estudo* - *modify (estudo)* - o que pode causar uma situação de *loop* infinito. Neste exemplo é possível verificar a utilização do atributo *saliency*, pois é uma regra prioritária:

```
rule "Betao NAO Auto-Compactavel"  
activation-group "autocompactavel"  
saliency 10  
no-loop
```

```

when
  estudo : Estudo ( Consistencia >= 0, Consistencia < 4 )
then
  SistemaPericial.frmSP.outputExplicacao("Betao não Auto-Compactável!");
  estudo.AutoCompactavel = false;
  modify(estudo);
end

```

### 5.5.3 Agrupamento de regras

Um outro atributo importante é o *activation-group*, o qual é utilizado para agrupar regras de modo a indicar explicitamente ao motor de inferência que apenas *uma* das regras do grupo pode disparar (mesmo que outras no mesmo grupo satisfaçam as condições).

A regra abaixo pertence ao *activation-group* "autocompactavel", tal como a regra acima, mas apenas *uma* delas pode disparar numa dada altura, dado que são antagónicas.

```

rule "Betao Auto-Compactavel"
activation-group "autocompactavel"
salience 10
no-loop
  when
    estudo : Estudo ( Consistencia == 4 )
  then
    SistemaPericial.frmSP.outputExplicacao("Betao Auto-compactavel.");
    estudo.AutoCompactavel = true;
    modify(estudo);
  end
end

```

O Drools dispõe ainda de outros mecanismos de resolução de conflitos, não tendo sido, no entanto, necessário usá-los.

## 5.6 Utilização do *Betacomp*

Neste ponto será demonstrado o funcionamento do *Betacomp* através de um exemplo de estudo de composição.

A interface do SP foi desenhada de forma a ser amigável para o utilizador nas suas escolhas, através do uso de *comboboxes*, *sliders* e *checkboxes*. Deste modo, é possível preencher todas as entradas

necessárias recorrendo apenas ao uso do rato e simultaneamente evitando o mais possível enganos ou a introdução de valores inválidos. Os campos a brancos são de *input* do utilizador e os amarelos de *outputs* do SP.

Para uma introdução à lógica seguida pelo SP, é apresentado na figura 5.6 um fluxograma que ilustra a sequência do raciocínio do sistema pericial para a composição de um betão. É possível constatar as semelhanças existentes com o fluxograma apresentado atrás no capítulo 2 (ver figura 2.4), dado que se tentou reproduzir o mais possível a lógica inerente ao processo da composição de betões de granulometria óptima [Lourenço et al., 2004].

### 5.6.1 Introdução dos *inputs* pelo utilizador

O utilizador começa por utilizar o SP através da interface do sistema pericial (ver figura 5.7), onde realiza a especificação do betão, isto é, quais as características esperadas do betão. As escolhas do utilizador serão marcadas a **negrito**. Uma especificação possível poderia ser, a título de exemplo:

- *Classe de exposição ambiental* - **XC1** (Ambiente seco ou permanentemente húmido);
- *Classe de resistência* - **C45/55** (a resistência mecânica necessária para o betão);
- *Classe de consistência* - **S3** (Trabalhabilidade do betão);
- *Dimensão máxima dos agregados* - **20 mm** (O tamanho limite do maior agregado, normalmente um tipo de brita).

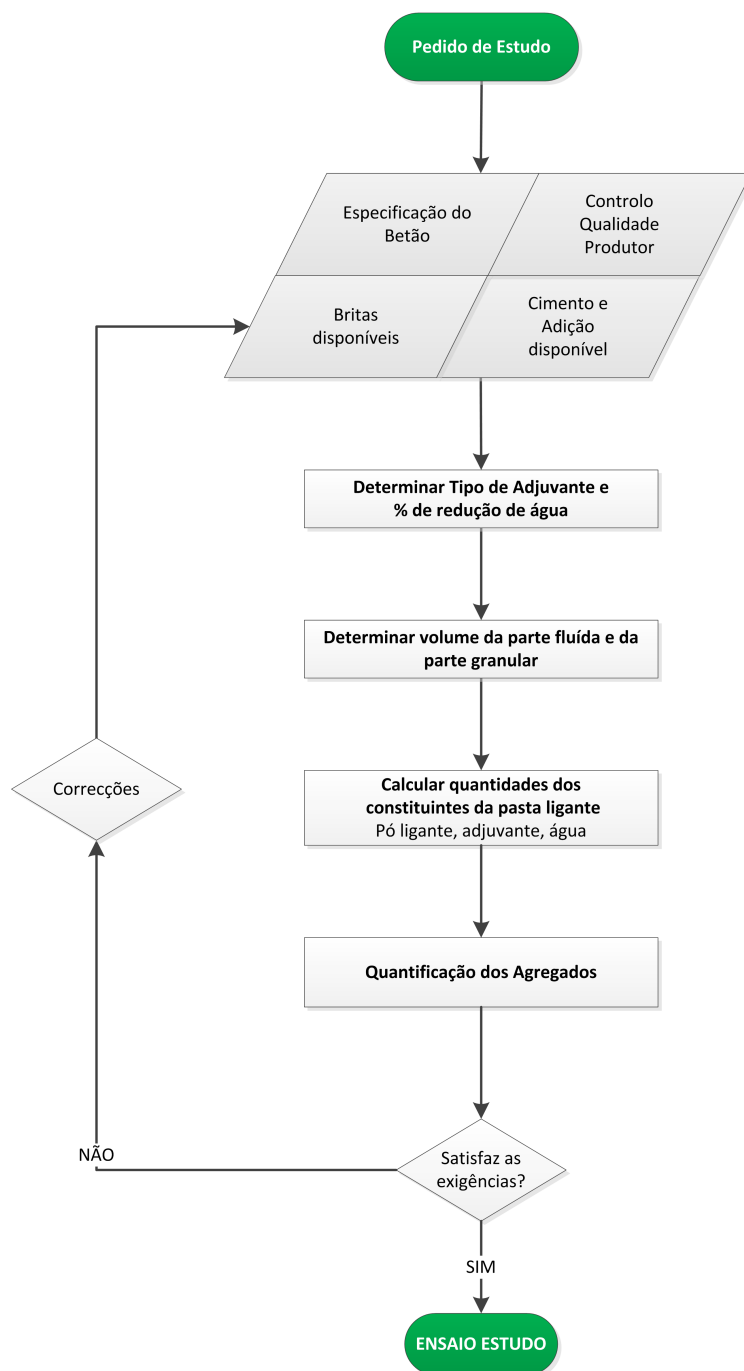
Em seguida, temos um parâmetro inovador, que não foi encontrado noutros sistemas de composição do betão. Com a indicação do *nível de qualidade do produtor do betão*, é feito um ajuste de compensação à resistência do betão a cumprir, proporcional à qualidade do produtor. Quanto menor a qualidade, mais resistência terá de ter. Vamos escolher o valor **Muito Bom**.

O passo seguinte é a escolha e caracterização dos constituintes disponíveis. Estes são:

- Cimento - **CEM II/A-L 42,5**;
- Adições - **Cinzas Volantes**;
- Britas - **Brita 1, Brita 2**.

Esta é outra inovação do sistema, pois permite que os produtores de betão indiquem os constituintes que já possuem para os poderem aproveitar, uma prática muito comum. No caso de não satisfazerem os requisitos da especificação, o sistema alertará para o facto. No anexo A é dada uma explicação mais completa sobre este tópico. É importante referir que a escolha dos constituintes disponíveis não é obrigatória. Caso não sejam indicados, o SP considera os que achar mais apropriados para o efeito.

Neste momento o SP tem todos os dados de que necessita para definir a composição do betão.



**Figura 5.6:** Fluxograma representativo do raciocínio do sistema pericial desenvolvido

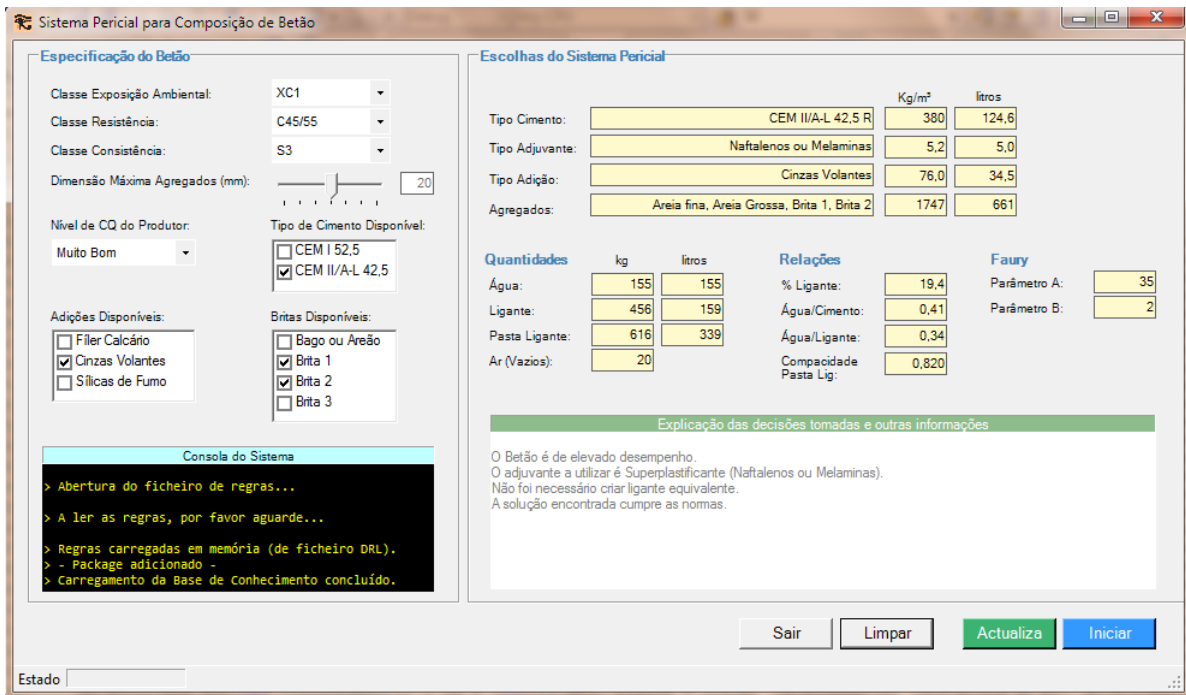


Figura 5.7: Resultados na interface do sistema pericial - *Betacom*

## 5.6.2 Inferência

A partir deste momento, o *Betacom* vai começar a processar os *inputs*. Os alertas e informações aparecem na caixa de texto *Explicação da composição e outras informações* no canto inferior direito da janela da interface e os valores definidos nas caixas a amarelo (*outputs* do sistema).

As verificações seguidas pelo SP estão delineadas nos passos seguintes:

1. verifica, com base na consistência especificada, se o betão é normal ou auto-compactável, dado que o último tem requisitos especiais e limitações;
2. se auto-compactável, valida se a dimensão máxima dos agregados está conforme, se não estiver alerta o utilizador; valida também se a brita não excede o tamanho máximo;
3. classifica o betão de acordo com a sua resistência, este pode ser:
  - betão corrente;
  - betão de elevado desempenho;
  - betão de elevadíssimo desempenho.
4. de acordo com o tipo de betão, define o tipo de adjuvante a utilizar;

5. definição da percentagem de redução de água necessária, com base no tipo de adjuvante, tendo em consideração a dimensão máxima dos agregados e a consistência desejada;
6. define percentagem de adjuvante necessária;
7. define volume do ar;
8. cálculo da compacidade da pasta ligante:
  - volume da parte fluída (água + adjuvante + ar);
  - volume da parte granular (cimento + adições + agregados).
9. Determinar ligante equivalente, com base na classe de exposição ambiental:
  - verifica se cimento disponível necessita de ligante equivalente;
  - verifica se a adição disponível pode ser utilizada;
  - escolha razão adição/cimento.
10. Determinação das quantidades dos constituintes (cimento, adição, adjuvante, água);
11. Determina número de agregados (3-5), com base no tipo de betão desejado (corrente, elevado desempenho, elevadíssimo desempenho, auto-compactável);
12. Determina variáveis de Faury, com base na classe de consistência desejada;
13. validação da normas EN206 e E464.

Neste momento as decisões do SP são apresentadas no ecrã, ou seja, os constituintes necessários para satisfazer os requisitos prévios, assim como as suas quantidades e alguns indicadores que definem características importantes do comportamento do betão, com a relação água/cimento ou a compacidade da pasta ligante. De igual modo aparecem as explicações que forem relevantes.

Os valores encontrados pelo SP foram:

- Constituintes
  - Tipo Cimento e sua quantidade: **CEM II/A-L 42,5 R - 380 kg**;
  - Tipo Adjuvante e sua quantidade: **Superplastificante (Melaminas) - 5,2 kg**;
  - Tipo Adição e sua quantidade: **Cinzas Volantes - 76 kg**;
  - Agregados e sua quantidade: **Areia fina, Areia Grossa, Brita 1, Brita 2 - 1747 kg**.
- Quantidades

- Água: **155** litros;
- Ligante: **159** litros;
- Pasta Ligante: **339** litros;
- Ar: **20** litros.

- **Relações**

- Percentagem de Ligante: **19,4**;
- Água / Cimento: **0,41**;
- Água / Ligante: **0,34**;
- Compacidade da Pasta Ligante: **0,468**.

- **Faury**

- Parâmetro A: **35**;
- Parâmetro B: **2**.

- **Explicações**

*Betão de elevado desempenho. Para ambiente seco ou permanentemente húmido. Não foi necessário criar ligante equivalente. O adjuvante a utilizar é Superplastificante (Naftalenos ou Melaminas). A solução encontrada cumpre as normas.*

### **5.6.3 Módulo de cálculo de composição de betão**

A partir deste momento, o controlo passa para o Módulo de cálculo de composição de betão, juntamente com os valores determinados pelo SP, como se pode ver na figura 5.8. São descritas as acções possíveis, pela ordem de apresentação:

#### **1. Betão e Pasta Ligante**

Neste primeiro ecrã são mostrados os dados relativos ao betão e à pasta ligante, onde podemos encontrar os valores sugeridos pelo SP e gráficos da compacidade e dos diferentes constituintes da pasta, para uma melhor compreensão da composição do betão. É também possível definir os valores em euros na grelha dos constituintes, o que será importante para a análise de custos (ver figura 5.11). No entanto, o seu preenchimento não é obrigatório.

#### **2. Agregados**

Após ter os campos necessários preenchidos, podemos passar ao painel dos *Agregados*. Aqui será preciso completar a grelha dos Agregados, apresentada na figura 5.9, os quais dependem dos

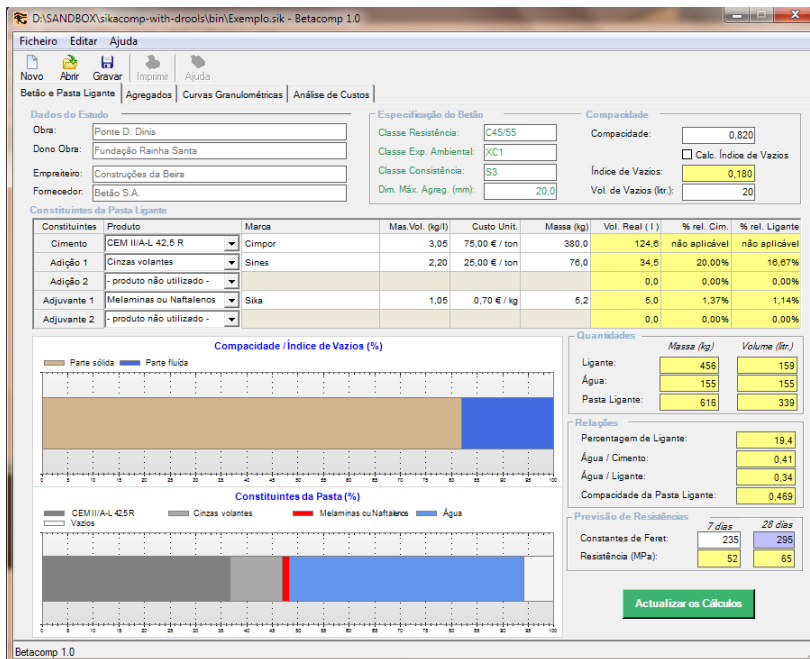


Figura 5.8: O ecrã inicial do módulo de cálculo de composição do *Betacomp*

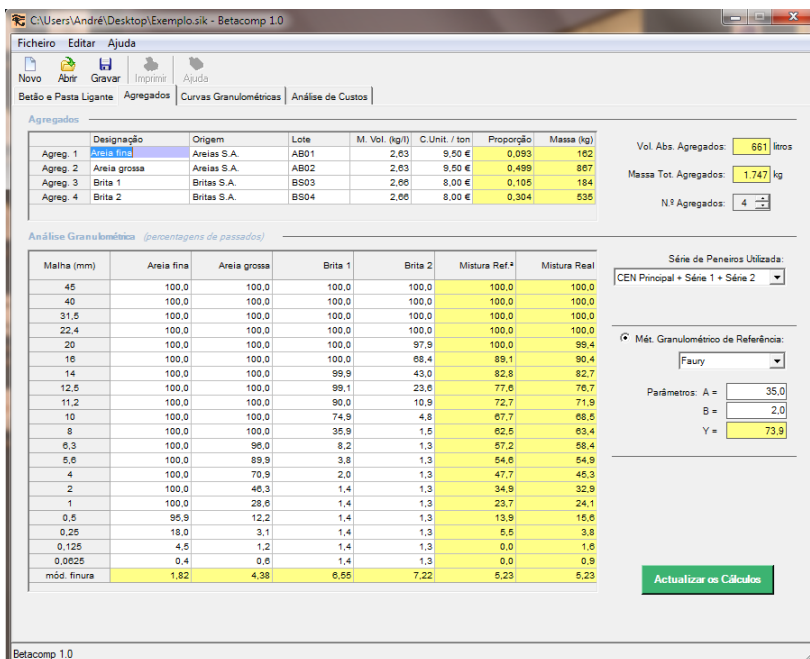


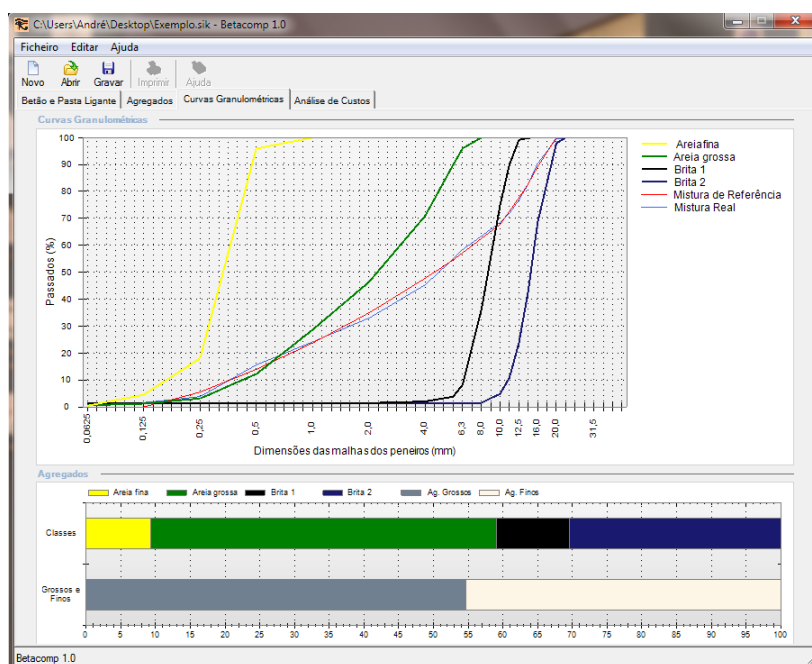
Figura 5.9: A análise granulométrica no *Betacomp*

materiais disponíveis localmente na obra. A grelha inferior, é a *Análise Granulométrica*, onde será necessário colocar a percentagem de material (agregados finos e grossos) que passaram nos peneiros.

Após estar preenchida, podemos utilizar o método de Faury, o qual necessita dos parâmetros *A* e *B* (previamente calculados pelo SP), para gerar os valores necessários para preencher as colunas da *Mistura de Referência* e da *Mistura Real*.

### 3. Curvas Granulométricas

No painel seguinte, é possível ver as *Curvas Granulométricas* com a representação da percentagem de passados de cada constituinte, e a mistura real (que junta todos os agregados) - ver figura 5.10. A mistura de referência serve aqui como uma indicação do objectivo a atingir, pelo que a mistura real deve estar o mais próxima possível desta, idealmente sobrepondo-se (o que na prática é muito difícil de acontecer na totalidade).



**Figura 5.10:** A análise gráfica das curvas granulométricas e dos agregados no *Betacomp*

### 4. Análise de Custos

Por fim chegamos à *Análise de Custos*, onde é possível visualizar o custo estimado do betão por  $m^3$  e o peso relativo de cada constituinte, como está patente na figura 5.11.

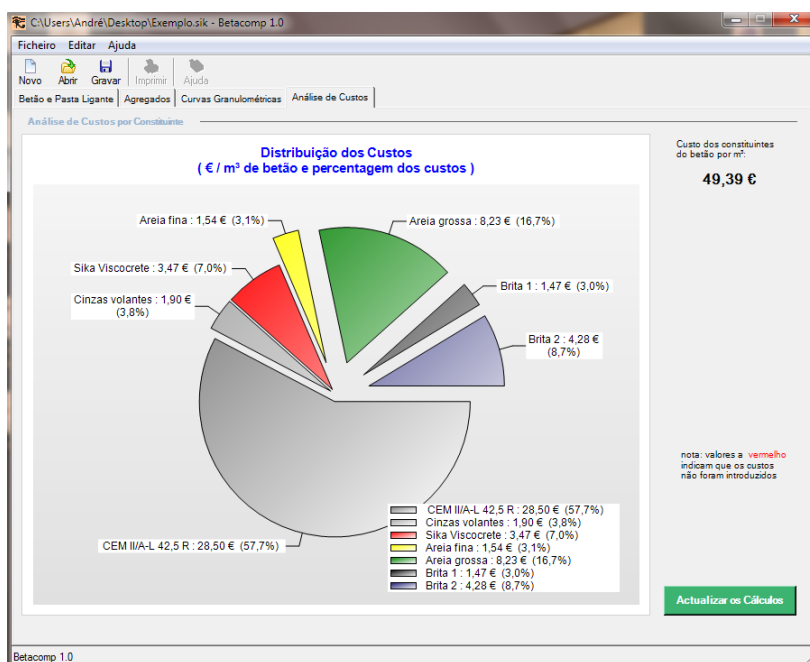


Figura 5.11: A análise de custos no *Betacomp*

## 5.7 Explicação das soluções

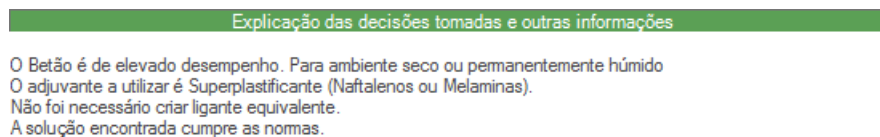
A explicação das soluções encontradas é um passo importante para a credibilidade e compreensão de qualquer sistema pericial, assim como um *feedback* positivo para o utilizador.

Esta foi uma questão que foi necessário resolver, pois o Drools não possui a funcionalidade de explicações de modo nativo. No entanto, foi criado um sistema de explicações que, pelo uso de regras simples, informa o utilizador na altura conveniente, sempre que há alguma decisão tomada pelo sistema.

Foi acrescentada a informação necessária nas classes de especificação do betão e numa classe criada só para o efeito, na interface do SP - a qual relembramos que foi escrita em *C#*. Todavia, o controlo das explicações é realizado na íntegra pelas regras Drools, pois são sempre estas que despoletam as invocações. Estas são as explicações relativas às conclusões relacionadas com as decisões do SP, as quais se dividem nas seguintes categorias:

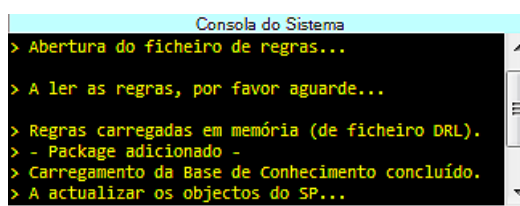
- Informações úteis (como qual o tipo de resistência do betão a elaborar ou o adjuvante a escolher);
- Correções a efectuar (se necessário);
- Resultados das validações das normas (se está conforme ou não);
- Outras validações.

A informação é mostrada ao utilizador na *Explicação das decisões tomadas e outras informações* na parte inferior da interface do SP, como se pode constatar na figura 5.12.



**Figura 5.12:** A *Explicação das decisões tomadas e outras informações* no *Betacomp*

A outro nível, foi criada uma *Consola de Sistema* no canto inferior esquerdo da interface do sistema pericial, onde são dadas explicações do funcionamento interno do SP, como se pode ver na figura 5.13, de modo a haver um melhor entendimento do mesmo.



**Figura 5.13:** A *Consola de Sistema* no *Betacomp*

## 5.8 Validação do sistema

Após o desenvolvimento do sistema, este foi testado pelo perito que forneceu o conhecimento pericial do *Betacomp* e pelo proponente.

O perito testou o programa prestando atenção às explicações e aos valores debitados pelo SP na realização de vários testes, com valores limite e diferentes cenários, tendo-se observado que o programa teve o comportamento esperado sem indicações especiais a reportar.

É importante referir que o sistema pericial inclui as suas próprias validações para que o betão esteja conforme, não só através da aplicação das limitações referidas pelo perito mas também pela obrigatoriedade da conformidade com as normas legais, mais especificamente a NP EN 206-1 e a E464. O facto

de os valores introduzidos pelo utilizador estarem limitados à partida, dado que todos os *inputs* são realizados através de componentes de escolha/selecção na interface, julgamos ser também um factor que condiciona situações não contempladas.

No caso de o betão não estar de acordo com o esperado, o utilizador é informado do facto através das explicações do sistema e pode efectuar as correcções.

As situações que o programa pode não suportar serão aquelas em que a especificação do betão está errada, ou seja, contém contradições. Nesse caso o utilizador é alertado acerca dos parâmetros que não estão a ser cumpridos.



## Capítulo 6

# Conclusões

*“En esta hora  
no niego que tuve tiempo,  
tiempo tuve,  
pero no tuve manos”*

—Pablo Neruda, *Las manos del día* (1968)

Não existe uma maneira única de compor um betão, o que por outras palavras significa que diferentes combinações dos constituintes e das suas proporções podem resultar em betões com características semelhantes e perfeitamente satisfatórias. Isto ilustra também o porquê de existirem diversos métodos e normas para composição de betão, e porque habitualmente cada perito segue uma metodologia um pouco diferente da dos seus colegas de profissão, fruto da sua experiência e também das suas preferências pessoais.

Por conseguinte, existe uma multiplicidade de combinações que podem resultar e fornecer um betão com as características desejadas.

No desenvolvimento do *Betacomp* foi adoptada uma destas metodologias com sucesso num sistema pericial, e mesmo considerando que é ainda um protótipo, tem uma componente de cálculo e apoio à decisão na composição do betão utilizada na indústria há já alguns anos na aplicação *Sikacomp 2.5*, também desenvolvida pelo proponente e sob a orientação técnica do Eng.º Jorge Lourenço, o perito consultado e co-orientador desta dissertação.

### 6.1 Resultados obtidos pela solução desenvolvida

A solução desenvolvida atingiu todos os objectivos inicialmente definidos, mais concretamente:

- foi criada uma base de conhecimento com a informação importante para o tema em causa, traduzindo a parte relevante do *know-how* de um perito;

- é um sistema de apoio à decisão que tem como base um sistema pericial que permite realizar estudos de composição de betão, desde os requisitos iniciais até ao ensaio em laboratório;
- a aplicação permite atingir rapidamente resultados, mesmo que o utilizador não possua conhecimentos aprofundados da matéria, podendo atingir um desempenho comparável ao de um perito (isto não significa necessariamente que se possa dispensar o perito);
- suporta os tipos de betão mais comuns, sendo aplicável numa grande parte dos cenários de obra;
- fornece explicações sobre as suas decisões, auxiliando a perceber as conclusões atingidas;
- tem uma vertente pedagógica, permitindo criar facilmente diferentes cenários e treinar os utilizadores na análise e experimentação da composição do betão.

## 6.2 Contribuições da solução desenvolvida

O *Betacomp* é uma solução completa para a definição da mistura do betão, dado que permite realizar os estudos recebendo do utilizador a especificação do betão e fornecendo no final uma composição completa e detalhada do betão a produzir em ensaios de laboratório.

A aplicação pode ser utilizada nos tipos mais comuns de betão, cobrindo uma extensa gama quando em comparação com os restantes sistemas analisados, que normalmente só consideram um ou dois tipos. As seguintes categorias são contempladas:

- betão corrente;
- betão de elevado desempenho;
- betão de elevadíssimo desempenho;
- betão auto-compactável.

A sua abordagem é bastante pragmática e tem em conta parâmetros novos, que habitualmente não são considerados neste tipo de *software*. Um deles é o nível do *controlo de qualidade do produtor de betão*, pois com esta indicação, é feito um ajuste de compensação à resistência do betão a cumprir, proporcional à qualidade do produtor - obtendo resultados mais ajustados à realidade.

Para além de considerar adições e adjuvantes na composição do betão, o que só uma minoria dos sistemas realizava, suporta as normas portuguesas e europeias obrigatórias por lei, o que com a exceção do *Sikacomp 2.5*, mais nenhum sistema suportava.

No sentido de se adaptar à realidade das centrais de betão, o sistema possui outra inovação, pois permite que os produtores de betão indiquem os constituintes que já possuem para os poderem aproveitar

(caso não haja impedimentos técnicos), uma prática muito comum e que permite eventualmente uma aceitação maior da aplicação, dado que reflecte a forma habitual de agir nos produtores.

O sistema desenvolvido é também um sistema moderno, que pode ser instalado em qualquer computador com sistema operativo *Microsoft Windows* actual. Embora possa parecer algo óbvio, a maioria dos sistemas analisados não pode ser instalada nos sistemas operativos que utilizamos hoje em dia.

### 6.3 Desenvolvimentos futuros

Embora o sistema se possa considerar razoavelmente completo, haveria um complemento que seria interessante ver em produção no *Betacomp*. Neste momento, o sistema conclui quando sugere a composição do betão para a realização de ensaios (ou seja, a passagem à prática em laboratório para fazer o betão). No entanto a prática normal exige que se façam correcções à composição do betão em função dos ensaios laboratoriais realizados, pois na manufactura do betão as equações nem sempre correspondem ao resultado prático - daí a epígrafe no capítulo 5.

Poderá portanto ser vantajoso adicionar à aplicação uma metodologia sistemática de correcções face ao resultado dos primeiros ensaios. Esta abordagem deverá ser feita com apoio na aplicação e nas seguintes recomendações, aqui genericamente mencionadas:

- Desvios nas consistências obtidas do betão, face às previsíveis;
- Correcções volumétricas (diferenças nas massas volúmicas);
- Desvios nas resistências reais face às esperadas.

Estes dados poderiam ser usados para alimentar novamente o sistema de forma a se obter uma nova composição do betão, cujas características sejam mais próximas das pretendidas.

Uma outra abordagem que poderia trazer uma mais-valia ao sistema seria dotá-lo de capacidade para lidar com conhecimento impreciso. Esta funcionalidade poderia ser útil, por exemplo, na indicação de um nível de confiança associado ao produtor, dado que é uma medida com alguma subjectividade inerente, logo um mecanismo que lidasse com incerteza poderia ser útil, como a aplicação de *fuzzy logic*.

A versão do Drools utilizada neste projecto (*Drools.NET*) é uma adaptação da versão Drools 3.0.6, o qual infelizmente não possui suporte nativo para este tipo de funcionalidade. A versão actual no entanto - Drools 5.x - já contempla este mecanismo, o que poderia trazer valor acrescentado a este trabalho, ao permitir outro tipo de cenários e raciocínio. Se considerarmos que o código da versão 3.x é facilmente adaptável ao da versão 5.x, esta parece ser uma opção exequível.



# Bibliografia

- Abdullahi, M., Al-Mattarneh, H., & Mohammed, B. (2009). A matlab program for diagnosis and adjustment of mix proportions of structural lightweight concrete. *European Journal of Scientific Research*, 106-123.
- Abdullahi, M., Al-Mattarneh, H. M. A., Hassan, M. H. A., & Mohammed, B. S. (2008). A review on expert systems for concrete mix design. *International Conference on Construction and Building Technology 2008*, 8.
- Advanced Computing Systems, C. for Research in. (2010). *Yaprolog*. <http://www.dcc.fc.up.pt/~vsc/Yap/>
- Akhras, G., & Foo, H. (1994). A knowledge-based system for selecting proportions for normal concrete. *Expert Systems with Applications*, 7(2), 323-335.
- Albert, L. (1989, December 19-21). Average case complexity analysis of rete pattern-match algorithm and average size of join in databases. *Foundations of Software Technology and Theoretical Computer Science*, 223-241.
- APEB - Associação Portuguesa das Empresas de Betão Pronto. (2008, Maio). *Guia para a utilização da norma np en 206-1 a especificação do betão*. PDF.
- APICe. (2011). *tuprolog*. <http://alice.unibo.it/xwiki/bin/view/Tuprolog/>
- Aitcin, P. (1998). *High-performance concrete* (T. . Francis, Ed.). E and FN Spon.
- Bali, M. (2009). *Drools jboss rules 5.0 developer's guide* (G. Singh, Ed.). Packt Publishing.
- Bay, Y., & Amirkhanian, S. (1994). Knowledge-based expert system for concrete mix design. *Journal of construction engineering and management, ASCE*, 120(2), 357-373.
- Brick, T. (2006, October). Ops5: A production rule system for expert systems [Computer software manual]. Department Of Computer Science & Engineering 384 Fitzpatrick Hall Notre Dame, In 46556 USA. [http://www.cse.nd.edu/courses/cse60431/www/BasicPage/lang\\_papers/OPS5.pdf](http://www.cse.nd.edu/courses/cse60431/www/BasicPage/lang_papers/OPS5.pdf)

- Brito, J. de, Branco, F. A., Thoft-Christensen, P., & Sørensen, J. D. (1997). An expert system for concrete bridge management. *Engineering Structures*, Vol. 19, No. 7, 519-526.
- Cohn, L. F., & Harris, R. A. (1990). *Knowledge based expert systems in transportation* (N. R. C. Transportation Research Board, Ed.). National Cooperative Highway Research Program.
- Corkill, D. D. (1991, September). Blackboard systems. *AI Expert*, 6 (9), 40-47.
- d3web. (2011, Julho). *Knowledge systems with d3web*. <http://www.d3web.de>
- EXSYS. (2012, Set). *Exsys inc - the expert system experts*. website. <http://www.exsys.com/>
- Forgy, C. L. (1981, Jan). Ops5 user's manual (Paper 2397 ed.) [Computer software manual]. <http://repository.cmu.edu/compsci/2397>
- Friedman-Hill, E. (1995). *Kapiclips 1.0*. online. <http://www.csee.umbc.edu/csee/research/kqml/software/kapiclips.shtml>
- Frijters, J. (2011). *Ikvm.net*. <http://www.ikvm.net/>
- Gensym. (2012). *G2 2011*. <http://www.gensym.com/product/G2>
- Giarratano, J. C., & Riley, G. D. (2005). *Expert systems principles and programming* (4th ed.). Thomson Course Technology.
- Giordano, M. (2003). *Webclips*. <http://clipsinterface.sourceforge.net/WebCLIPS/wchome.htm>
- Hecht-Nielsen, R. (1989). Theory of the backpropagation neural network. In *Neural networks, 1989. ijcnn., international joint conference on* (Vol. 1, p. 593-605).
- Hodroj, A. (2011). *Prolog.net*. <http://prolog.hodroj.net/>
- Jackson, P. (1999). *Introduction to expert systems* (3rd ed.). Addison-Wesley.
- Ji-Zong, W., Hong-Guang, N., & Jin-Yun, H. (1999). The application of automatic acquisition of knowledge to mix design of concrete. *Cement and Concrete Research* 29, 1875–1880.
- Kaetzel, L. J., & Clifton, J. R. (1991). *Expert/knowledge-based systems for cement and concrete: State-of-the-art report* (Tech. Rep.). National Research Council Washington, D.C.
- Laboratório Nacional de Engenharia Civil, L. (2012, Set). *Matwin - gestão de ensaios de materiais*. web. <http://www.lnec.pt/servicos/software/matwin>

- LNEC. (2007). *Lnec e 464 specification: Concrete. prescriptive methodology for a design working life of 50 and of 100 years under the environmental exposure* (Tech. Rep.). Author.
- Logic Programming Associates, L. (2012, September). *Winprolog*. <http://www.lpa.co.uk/>
- Lomborg, B. (2001). *The skeptical environmentalist: measuring the real state of the world* (C. U. Press, Ed.). Press Syndicate of the University of Cambridge.
- Lourenço, J., & Coutinho, J. (1986). *O cálculo automático no projecto de composição de betões (métodos de bolomey e de faury)* (Tech. Rep.). Comissão de Coordenação da Região Centro - Ministério do Plano e da Administração do Território.
- Lourenço, J., Júlio, E., & Maranhã, P. (2004). *Betões de agregados leves de argila expandida* (Tech. Rep.). Associação Portuguesa das Empresas de Betão Pronto.
- Malasri, S., & Maldonado, S. (1988a). Concrete mix designer. *Computer application in concrete technology, ACI, 111-03*, 33-42.
- Malasri, S., & Maldonado, S. (1988b). Concrete mix designer. *American Concrete Institute, Detroit MI, Special Publication, 111*.
- Mathworks, U., Massachusetts. (2012, Set). *Matlab*. Web. <http://www.mathworks.com/products/matlab/>
- Merritt, D. (1989). *Building expert systems in prolog* (Springer-Verlag, Ed.). Amzi! inc.
- Minsky, M. (1968). *Semantic information processing* (MIT, Ed.). MIT Press.
- Minsky, M. (1974). A framework for representing knowledge. *MIT-AI Laboratory Memo, 306*.
- Orchard, R. A. (1998). *Fuzzyclips version 6.04a user's guide*.
- Poole, C., Terrell, J., & Busoli, S. (2007). *Nunit 2.0*. <http://www.nunit.org/>
- Proctor, M. (2010). *Drools business logic integration platform*. [http://planet.jboss.org/post/jug\\_lille\\_video\\_s\\_drools\\_and\\_drools\\_planner\\_plus\\_other\\_presentations](http://planet.jboss.org/post/jug_lille_video_s_drools_and_drools_planner_plus_other_presentations)
- Proctor, M., Neale, M., Lin, P., & Frandsen, M. (2005). *Drools reference manual 3.0.6* [Computer software manual].
- Ramos, C. (2009). *Representação do conhecimento, apontamentos para acompanhamento das aulas de sistemas baseados em conhecimento do mestrado em engenharia informática do isep*. Instituto Superior de Engenharia do Porto.

- Rattanaprteep, C., & Chittayasothorn, S. (2007, sept.). A frame-based object relational expert database system. In *Africon 2007* (p. 1 -7).
- Riley, G. D. (2012, Setembro). *Clips a tool for building expert systems*. <http://clipsrules.sourceforge.net/WhatIsCLIPS.html>
- Salam, M., Islam, M., & Rashid, M. (2012). An expert system for mix design of brick aggregate concrete. *International Journal of Mechanical and Materials Engineering (IJMME)*, 7(1), 54–63.
- Schooten, B. van. (2006). *Yprolog*. <http://wwwhome.ewi.utwente.nl/~schooten/yprolog/>
- Seren, K. (1988). An expert system for choosing the type of ready mix concrete. *The Nordic Concrete Federation, Finland*, 7.
- Shaeffer, R. E. (1992). *Reinforced concrete: preliminary design for architects and builders*. McGraw-Hill.
- Sika Portugal SA. (1999, Novembro). *Sika breviário do betão* (Sika - Indústria Química, SA ed.; Tech. Rep.). Rua de Gondarém, 113 Apartado 2768 4401-601 Vila Nova de Gaia.
- Sika Portugal, S. (2007). *Sikacomp 2.5 cálculo de composição de betão* [Computer software manual].
- Smith, L. (1987). *Interim report on comix: An expert system for concrete mix design* (Tech. Rep. No. Report No. M4.87/1). Central Laboratories, New Zealand.
- Sourceforge. (2006). *Drools.net*. <http://sourceforge.net/projects/drooldotnet/>
- Strauss, M. (2007). *Jess - the java expert system shell* [Computer software manual].
- Wicklunf, G. A., & Roth, R. M. (1987). *Expert system implementations for business applications*. University of Iowa, College of Business Administration.
- Williams, T., Khajuria, A., & Balaguru, P. (1991, May). A knowledge-based expert system for quality assurance of concrete. *Proceedings of the Seventh Conference on Computing in Civil Engineering, ASCI, New York, NY*.
- Zain, M. F. M., Islam, M. N., & Basri, I. H. (2005). An expert system for mix design of high performance concrete. *Advances in Engineering Software, Volume 36, Issue 5*, Pages 325-337.

# **Anexos**



## Apêndice A

# Questões de ordem prática e tomada de opções

Conforme foi afirmado nas conclusões, não existe uma maneira única de compor um betão, o que significa que com diferentes combinações dos seus constituintes e respectivas proporções podemos obter betões com características idênticas e igualmente satisfatórias.

Logo há uma multiplicidade de combinações que podem resultar e fornecer um betão com as características desejadas.

Esta noção é fulcral, pois permite-nos entender algumas das opções realizadas no desenvolvimento deste sistema pericial. Num cenário real numa central de betão pronto, onde é realizada a maioria dos betões hoje em dia, existem silos que albergam grandes quantidades de material, cada qual com um constituinte. Dado que a quantidade armazenada é muito grande, após a realização de um betão para uma obra específica, a central irá manter as toneladas que ainda sobraram no silo desse mesmo constituinte para a obra seguinte. Só em situações muito excepcionais poderá ocorrer o esvaziamento integral de um silo cheio ou semi-cheio para albergar um tipo diferente de cimento ou adição, se é forçosamente necessário, por exemplo, utilizar um determinado tipo de cimento para garantir as especificações do betão.

Assim, o cenário habitual numa central de betão é encontrarmos silos com um determinado tipo de cimento e outros com um determinado tipo de adição para criar um betão, os quais deverão ser utilizados na feitura do betão encomendado.

Isto explica o facto de o SP desenvolvido questionar o utilizador sobre quais os cimentos e adições disponíveis. As circunstâncias são idênticas em relação aos agregados grossos, como as britas. Este constituiu um ponto importante no desenvolvimento deste software, pois trata-se de uma aproximação pragmática que reflecte o que acontece nos cenários reais de produção de betão.

Uma aproximação diferente poderia ser simplesmente sugerir um cimento considerado ideal (mesmo

que marginalmente) mas que seria muito mais oneroso, quer em termos logísticos, quer em termos de valor corrente no mercado. Essa seria possivelmente uma abordagem mais teórica e, de certo modo, mais desligada da realidade.

O cimento é em regra mais caro que as adições, pelo que normalmente há interesse em aumentar a quantidade de adição na mistura de modo a conseguir-se um ligante equivalente a um menor custo. A exceção aplica-se às sílicas de fumo, utilizadas para aumentar a resistência do betão, que são especialmente dispendiosas.

## Apêndice B

# Regras de produção do *Betacomp*

Ficheiro <*DroolsRules.drl*>

```
package SistemaPericial

import SistemaPericial.Classes

/*****
*****
***** CONFORMIDADE COM AS NORMAS NACIONAIS */

/*----- Teor mínimo de ar -----*/

rule "Teor Min Ar Obrigatorio"
salience -10
//no-loop
when
    estudo: Estudo( ExpAmbiental >= 13, ExpAmbiental <= 15) #XF2 a XF4
then
    SistemaPericial.frmSP.outputExplicacao("Teor mínimo de ar obrigatório de 4% pois
        é um betão XF.");
    estudo.TeorMinAr = 4;
    // SistemaPericial.frmSP.debugResult("C15/20");
end

rule "Teor Min Ar NAO Obrigatorio"
salience -10
//no-loop
```

```

when
    estudo: Estudo( ExpAmbiental < 13) or Estudo( ExpAmbiental > 15) #XF2 a XF4
then
    estudo.TeorMinAr = 0;
    // modify (estudo);
    // SistemaPericial.frmSP.debugResult("C15/20");
end

/*----- CIMENTO -----*/

rule "Cimento Forte"
saliency -10
when
    #cimento : Cimento( Strength >= 1, Strength <= 3)
    cimento : Cimento( SubTipo matches "A" )
then
    #SistemaPericial.frmSP.outputExplicacao("Cimentos encontrados:\n");
    SistemaPericial.frmSP.outputExplicacao(cimento.DescCompletaCimento());
end

rule "Mostra Explicacoes ExpAmb"
saliency -5
no-loop
when
    Estudo( id : ExpAmbiental ) and ea : ClasseExpAmb (Id == id)
then
    SistemaPericial.frmSP.outputExplicacao(ea.Ambiente);
end

rule "Mostra Explicacoes Adjuvante"
saliency -5
no-loop
when
    Estudo( id : Adjuvante ) and adjuv : Adjuvante (Id == id)
then
    SistemaPericial.frmSP.outputExplicacao(adjuv.Explicacao);
end

rule "Mostra Explicacoes Adicao"
saliency -5
no-loop
when
    Estudo( id : Adicao ) and adicao : Adicao (Id == id)
then

```

```

        SistemaPericial.frmSP.outputExplicacao(adicao.Explicacao);
end

/*----- EXPOSICAO AMBIENTAL e CLASSE DE RESISTÊNCIA MÍNIMA -----*/

rule "X0"
saliency -10
  when
    estudo : Estudo ( ExpAmbiental == 1 ) #X0
    expamb : ClasseExpAmb ( Id == 1)
  then
    estudo.ResistenciaMinima = 1; #C12/15
  end

rule "ResistMin XC1 e XC2"
saliency -10
  when
    estudo : Estudo ( ExpAmbiental >= 2, ExpAmbiental <= 3 ) #XC1
  then
    estudo.ResistenciaMinima = 3; #C25/30
  end

rule "ResistMin XC3"
saliency -10
  when
    estudo : Estudo ( ExpAmbiental == 4 ) #XC1
  then
    estudo.ResistenciaMinima = 4; #C30/37
  end

rule "ResistMin XC4"
saliency -10
  when
    estudo : Estudo ( ExpAmbiental == 5 ) #XC1
  then
    estudo.ResistenciaMinima = 4; #C30/37
  end
//-----
rule "ResistMin XD1 e XD2"
saliency -10
  when
    estudo : Estudo ( ExpAmbiental >= 6, ExpAmbiental <= 7 ) #XD1 ou XD2
  then
    estudo.ResistenciaMinima = 4; #C30/37

```

```

end

rule "ResistMin XD3"
saliency -10
  when
    estudo : Estudo ( ExpAmbiental == 8 ) #XD1 ou XD2
  then
    estudo.ResistenciaMinima = 5; #C35/45
  end
//-----
rule "ResistMin XS1 e XS2"
saliency -10
  when
    estudo : Estudo ( ExpAmbiental >= 9, ExpAmbiental <= 10 ) #XD1 ou XD2
  then
    estudo.ResistenciaMinima = 4; #C30/37
  end
end

rule "ResistMin XS3"
saliency -10
  when
    estudo : Estudo ( ExpAmbiental == 11 ) #XD1 ou XD2
  then
    estudo.ResistenciaMinima = 5; #C35/45
  end
end

//-----
rule "ResistMin XFX"
saliency -10
  //no-loop
  when
    estudo: Estudo( ExpAmbiental >= 12, ExpAmbiental <= 15) #XF1 a XF4
  then
    estudo.ResistenciaMinima = 4; #C30/37
  end
end
//-----
rule "ResistMin XA1"
saliency -10
  when
    estudo : Estudo ( ExpAmbiental == 16 ) #XD1 ou XD2
  then
    estudo.ResistenciaMinima = 4; #C30/37
  end
end

```

```

rule "ResistMin XA2 e XA3"
saliency -10
  when
    estudo : Estudo ( ExpAmbiental >= 17, ExpAmbiental <= 18 ) #XA2 ou XA3
  then
    estudo.ResistenciaMinima = 5; #C35/45
  end

rule "Quantidade minima de cimento"
saliency -10
  when
    estudo : Estudo ( QtdCimento < QtdCimentoMinima )
  then
    SistemaPericial.frmSP.outputExplicacao("A quantidade de cimento não é suficiente
      . Por favor refaça a sua escolha"); #C35/45
    estudo.FlagNormas = false;
  end

rule "Quantidade minima de ar"
saliency -10
  when
    estudo : Estudo ( VolumeAr < TeorMinAr )
  then
    SistemaPericial.frmSP.outputExplicacao("A quantidade de ar não é suficiente. Por
      favor refaça a sua escolha"); #C35/45
    estudo.FlagNormas = false;
  end

rule "Resistencia Minima"
saliency -10
  when
    estudo : Estudo ( Resistencia < ResistenciaMinima )
  then
    SistemaPericial.frmSP.outputExplicacao("A resistência do betão não é suficiente.
      Por favor refaça a sua escolha"); #C35/45
    estudo.FlagNormas = false;
  end

rule "Normas nao conforme"
activation-group "normas"
no-loop
saliency -10
  when

```

```

    estudo : Estudo(FlagNormas == false)
  then
    SistemaPericial.frmSP.outputExplicacao("O betão não está conforme as normas. Por
      favor refaça a sua escolha");
  end

rule "Normas OK"
activation-group "normas"
no-loop
saliency -10
  when
    estudo : Estudo(FlagNormas == true)
  then
    SistemaPericial.frmSP.outputExplicacao("A solução encontrada cumpre as normas.
      Por favor refaça a sua escolha");
  end

end

/*****
*****
***** BETÃO AUTO-COMPACTÁVEL */

rule "Betao NAO Auto-Compactavel"
activation-group "autocompactavel"
// saliency 10
no-loop
  when
    estudo : Estudo ( Consistencia >= 0, Consistencia < 4 )
  then
    SistemaPericial.frmSP.outputExplicacao("Betao NAO AC!");
    estudo.AutoCompactavel = false;
    modify(estudo);
    // if (exists(BetaoAutoCompactavel())) retract (BetaoAutoCompactavel);
  end

end

rule "Betao Auto-Compactavel"
activation-group "autocompactavel"
// saliency 10
no-loop
  when
    estudo : Estudo ( Consistencia == 4 ) # Consistencia SF2
  then
    SistemaPericial.frmSP.outputExplicacao("Betao Auto-compactavel.");
    estudo.AutoCompactavel = true;
    estudo.QtdCimentoMinima = 550;
  end

```

```

    modify( estudo );
    // if ( not exists( BetaoAutoCompactavel() ) ) assert( new BetaoAutoCompactavel() );
end

```

```

rule "DMax muito grande para AC"
saliency 10
no-loop
when
    estudo : Estudo ( DMax > 16, AutoCompactavel == true ) # DMax maior que 16mm
    // exists ( BetaoAutoCompactavel() )
then
    SistemaPericial.frmSP.outputExplicacao("O Betao Auto-compactavel nao suporta
        agregados superiores a 16mm. Por favor refaca a sua escolha.");
end

```

```

/*****
*****
***** DESEMPENHO DESEJADO PARA O BETÃO */

```

```

rule "Betao Corrente"
// activation-group "desempenho_betao"
// saliency 8
//no-loop
when
    estudo : Estudo ( Resistencia >= 1, Resistencia <= 6 ) # C8/10 a C30/37
then
    estudo.DesempenhoBetao = 0; # Betão Corrente
    // estudo.Adjuvante = 0;
    // modify( estudo );
end

```

```

rule "Betao Elevado Desempenho"
// activation-group "desempenho_betao"
// saliency 8
//no-loop
when
    estudo : Estudo ( Resistencia >= 7, Resistencia <= 11 ) # C35/45 a C55/67
then
    estudo.DesempenhoBetao = 1; # Betão Elevado Desempenho
    // modify( estudo );
end

```

```

rule "Betao Elevadissimo Desempenho"

```

```

// activation_group "desempenho_betao"
// salience 8
//no-loop
when
    estudo : Estudo ( Resistencia >= 12, Resistencia <= 16 ) # C60/75 a C100/105
then
    estudo.DesempenhoBetao = 2; # Betão Elevadíssimo Desempenho
    //modify(estudo);
end

```

```

/*****
*****
***** Aditivos

```

```

rule "Superplastificante"
// activation_group "adjuvante"
// salience 7
//no-loop
when
    estudo: Estudo(DesempenhoBetao == 0)
then
    SistemaPericial.frmSP.outputExplicacao(" Superplastificante 0.");
    estudo.Adjuvante = 0;
    //modify(estudo);
end

```

```

rule "Superplastificante 2"
// salience 7
//no-loop
when
    estudo: Estudo(DesempenhoBetao == 1)
then
    SistemaPericial.frmSP.outputExplicacao(" Superplastificante 1.");
    estudo.Adjuvante = 1;
    //modify(estudo);
end

```

```

rule "Superplastificante 3G"
// salience 7
//no-loop

```

```

when
    estudo: Estudo(AutoCompactavel == false)
    estudo: Estudo(DesempenhoBetao == 2)
then
    SistemaPericial.frmSP.outputExplicacao(" Superplastificante 3G");
    estudo.Adjuvante = 2;
    //modify(estudo);
end

rule "Superplastificante 3G e Inc Visc"
//saliency 7
//no-loop
when
    estudo : Estudo(AutoCompactavel == true)
    estudo : Estudo(DesempenhoBetao == 2)
then
    SistemaPericial.frmSP.outputExplicacao(" Superplastificante 3G com Inc. Visc.");
    estudo.Adjuvante = 3;
    //modify(estudo);
end

/*----- Volume de Ar -----*/

rule "Volume de Ar DMax 8mm"
activation-group "volume_ar"
when
    estudo : Estudo(DMax == 8)
then
    estudo.VolumeAr = 30;
    modify(estudo);
end

rule "Volume de Ar DMax 12 e 16 mm"
activation-group "volume_ar"
when
    estudo : ( Estudo(DMax == 12) or Estudo(DMax == 16) )
then
    estudo.VolumeAr = 25;
    modify(estudo);
end

rule "Volume de Ar DMax 20mm"
activation-group "volume_ar"
when

```

```

    estudo : Estudo(DMax == 20)
  then
    estudo.VolumeAr = 20;
    modify(estudo);
end

```

```

rule "Volume de Ar DMax 22mm"
activation-group "volume_ar"
  when
    estudo : Estudo(DMax == 22)
  then
    estudo.VolumeAr = 15;
    modify(estudo);
end

```

```

rule "Volume de Ar DMax 25 e 32 mm"
activation-group "volume_ar"
  when
    estudo : ( Estudo(DMax == 25) or Estudo(DMax == 32) )
  then
    estudo.VolumeAr = 10;
    modify(estudo);
end

```

/\*\*\*\*\*\*

## Reducao da Agua

\*\*\*\*\*/

```

rule "Reducao Agua 10"
  when
    adjuvante : Adjuvante(Id == 0)
    estudo : Estudo()
    // estudo : Estudo( Adjuvante == adjuvante )
  then
    estudo.ReducaoAgua = 10;
end

```

```

rule "Reducao Agua 20"
  when

```

```

    adjuvante : Adjuvante(Id == 1)
  then
    estudo.ReducaoAgua = 20;
end

rule "Reducao Agua 30"
  when
    adjuvante : ( Adjuvante(Id == 2) or Adjuvante(Id == 3) )
  then
    estudo.ReducaoAgua = 30;
end

rule "Percentagem Adjuvante 10"
  when
    estudo : Estudo(ReducaoAgua == 10)
  then
    estudo.PercAdjuvCimento = 0.50;
    modify(estudo);
end

rule "Percentagem Adjuvante 20"
  when
    estudo : Estudo(ReducaoAgua == 20)
  then
    estudo.PercAdjuvCimento = 1.50;
    modify(estudo);
end

rule "Percentagem Adjuvante 30"
  when
    estudo : Estudo(ReducaoAgua == 30)
  then
    estudo.PercAdjuvCimento = 2.20;
    modify(estudo);
end

rule "Calculo Parte Fluida e Granular"
no-loop
  when
    estudo : Estudo(VolumeAr > 0, PercAdjuvCimento > 0, Adjuvante > 0)
  then
    estudo.CalculoParteFluida();
    modify(estudo);
end

```

```

rule "Volume Agua"
no-loop
  when
    estudo : Estudo( QtdCimento > 0, QtdAdicao > 0, QtdAdjuvante > 0 )
  then
    estudo.CalculoVolAgua();
    modify(estudo);
  end
end

```

```

rule "Calculo de Quantidades"
no-loop
  when
    estudo : Estudo( VolParteGranular > 0 )
  then
    estudo.CalculoQtdConstituintes();
    modify(estudo);
  end
end

```

```

rule "Calculo de Quantidades"
no-loop
  when
    estudo : Estudo( VolParteGranular > 0 )
  then
    estudo.CalculoQtdConstituintes();
    modify(estudo);
  end
end

```

```

rule "Sem Ligante Equivalente"
no-loop
  when
    estudo : ( Estudo ( ExpAmbiental >= 2, ExpAmbiental <= 5 ) or Estudo (
      ExpAmbiental >= 12, ExpAmbiental <= 15 ) )
    #Exp. Ambiental XC, XF
  then
    estudo.LiganteEquiv = false;
    SistemaPericial.frmSP.outputExplicacao("Não foi necessário criar ligante
      equivalente.");
    modify(estudo);
  end
end

```

```

rule "Com Ligante Equivalente"
no-loop
  when
    estudo : ( Estudo ( ExpAmbiental >= 6, ExpAmbiental <= 11 ) or Estudo (
      ExpAmbiental >= 16, ExpAmbiental <= 18 ) )
    #Exp. Ambiental XC, XF
  then
    estudo.LiganteEquiv = true;
    SistemaPericial.frmSP.outputExplicacao("É necessário criar um ligante
      equivalente a CEM IV/A(v) ou CEM IV/B(v).");
    modify(estudo);
end

```

/\*\*\*\*\*\*

#### Parâmetros de Faury

\*\*\*\*\*/

```

rule "Faury S2"
  when
    estudo : Estudo(Consistencia == 1) # S2
  then
    estudo.FauryA = 29;
    estudo.FauryB = 1.5;
    modify(estudo);
end

```

```

rule "Faury S3"
  when
    estudo : Estudo(Consistencia == 2) # S3
  then
    estudo.FauryA = 35;
    estudo.FauryB = 2;
    modify(estudo);
end

```

```

rule "Faury S4"
  when
    estudo : Estudo(Consistencia == 3) # S4
  then

```

```
    estudo.FauryA = 41;
    estudo.FauryB = 2;
    modify(estudo);
end

rule "Faury SF2"
  when
    estudo : Estudo(Consistencia == 4) # SF2 Auto-Compactavel
  then
    estudo.FauryA = 44;
    estudo.FauryB = 2;
    modify(estudo);
  end
end
```

## Apêndice C

# Listagem parcial das explicações do sistema

É apresentada a título de exemplo uma das classes utilizadas para as explicações do sistema, neste caso as classes de exposição ambiental do betão. As descrições são invocadas a partir das regras de produção para produzir as explicações.

```
public static class ListaClassesExpAmbiental
{
    public struct X0 //Sem risco de corrosão ou ataque
    {
        public const string TipoExposicao = "Sem risco de corrosão ou ataque";
        public const string Classe = "X0";
        public const string Ambiente = "Para betão não armado e sem metais embebidos:
            todas as exposições , excepto ao gelo/degelo , à abrasão ou ao ataque químico.
            Para betão armado ou com metais embebidos: ambiente muito seco";
        public const string Exemplo = "Betão no interior de edifícios com muito pouca
            humidade do ar";
    }

    public struct XC //Corrosão induzida por carbonatação
    {
        public const string TipoExposicao = "Corrosão induzida por carbonatação";

        public const string Classe1 = "XC1";
        public const string Ambiente1 = "Seco ou permanentemente húmido";
        public const string Exemplo1 = "Betão no interior de edifícios com baixa
            humidade do ar; Betão permanentemente submerso em água";
    }
}
```

```

public const string Classe2 = "XC2";
public const string Ambiente2 = "Húmido, raramente seco";
public const string Exemplo2 = "Superfícies de betão sujeito a longos períodos
    de contacto com água; Muitas fundações";

public const string Classe3 = "XC3";
public const string Ambiente3 = "Moderadamente húmido";
public const string Exemplo3 = "Betão no interior de edifícios com moderada ou
    elevada humidade do ar; Betão no exterior protegido da chuva";

public const string Classe4 = "XC4";
public const string Ambiente4 = "Ciclicamente húmido e seco";
public const string Exemplo4 = "Superfícies de betão sujeitas ao contacto com a
    água, fora do âmbito da classe XC2";
}

public struct XD //Corrosão induzida por cloretos não proveniente da água do mar
{
    public const string TipoExposicao = "Corrosão induzida por cloretos não
        proveniente da água do mar";

    public const string Classe1 = "XD1";
    public const string Ambiente1 = "Moderadamente húmido";
    public const string Exemplo1 = "Superfícies de betão expostas a cloretos
        transportados pelo ar";

    public const string Classe2 = "XD2";
    public const string Ambiente2 = "Húmido, raramente seco";
    public const string Exemplo2 = "Piscinas; Betão exposto a águas industriais
        contendo cloretos";

    public const string Classe3 = "XD3";
    public const string Ambiente3 = "Ciclicamente húmido e seco";
    public const string Exemplo3 = "Partes de pontes expostas a salpicos de água
        contendo cloretos; Pavimentos; Lajes de parques de estacionamento de
        automóveis";
}

public struct XS //Corrosão induzida por cloretos da água do mar
{
    public const string TipoExposicao = "Corrosão induzida por cloretos da água do
        mar";
}

```

```

public const string Classe1 = "XS1";
public const string Ambiente1 = "Ar transportando sais marinhos mas sem contacto
    directo com a água do mar";
public const string Exemplo1 = "Estruturas na zona costeira ou na sua
    proximidade";

public const string Classe2 = "XS2";
public const string Ambiente2 = "Submersão permanente";
public const string Exemplo2 = "Partes de estruturas marítimas";

public const string Classe3 = "XS3";
public const string Ambiente3 = "Zonas de marés, de rebentação ou de salpicos";
public const string Exemplo3 = "Partes de estruturas marítimas";
}

public struct XF //Ataque pelo gelo/degelo com ou sem produtos descongelantes
{
    public const string TipoExposicao = "Ataque pelo gelo/degelo com ou sem produtos
        descongelantes";

    public const string Classe1 = "XF1";
    public const string Ambiente1 = "Moderadamente saturado de água, sem produtos
        descongelantes";
    public const string Exemplo1 = "Superfícies verticais de betão expostas à chuva
        e ao gelo";

    public const string Classe2 = "XF2";
    public const string Ambiente2 = "Moderadamente saturado de água, com produtos
        descongelantes";
    public const string Exemplo2 = "Superfícies verticais de betão de estruturas
        rodoviárias expostas ao gelo e a produtos descongelantes transportados pelo
        ar";

    public const string Classe3 = "XF3";
    public const string Ambiente3 = "Fortemente saturado, sem produtos
        descongelantes";
    public const string Exemplo3 = "Superfícies horizontais de betão expostas à
        chuva e ao gelo";

    public const string Classe4 = "XF4";
    public const string Ambiente4 = "Fortemente saturado, com produtos
        descongelantes";
    public const string Exemplo4 = "Estradas e tabuleiros de pontes expostos a
        produtos descongelantes; Superfícies de betão expostas ao gelo e a salpicos

```

```

        de água contendo produtos descongelantes; Zona das estruturas marítimas
        expostas à rebentação e ao gelo";
    }

public struct XA //Ataque químico
{
    public const string TipoExposicao = "Ataque químico";

    public const string Classe1 = "XA1";
    public const string Ambiente1 = "Ligeiramente agressivo";
    public const string Exemplo1 = "Água do mar. Solos naturais e águas subterrâneas
        contendo agentes químicos agressivos para o betão e para os elementos
        metálicos embebidos.";

    public const string Classe2 = "XA2";
    public const string Ambiente2 = "Moderadamente agressivo";
    public const string Exemplo2 = "Água do mar. Solos naturais e águas subterrâneas
        contendo agentes químicos agressivos para o betão e para os elementos
        metálicos embebidos.";

    public const string Classe3 = "XA3";
    public const string Ambiente3 = "Fortemente agressivo";
    public const string Exemplo3 = "Água do mar. Solos naturais e águas subterrâneas
        contendo agentes químicos agressivos para o betão e para os elementos
        metálicos embebidos.";
}
}
}

```

